FINAL REPORT

ON

"ADDED SCOPE"

TO

CONTRACT NAS8-34337

BETA EXPERIMENT

APRIL 30, 1983

PREPARED BY:

APPLIED RESEARCH, INC.
131 LONGWOOD DRIVE
P. O. BOX 194
HUNTSVILLE, ALABAMA 35804

Applied Research, Inc.

P. O. Box 194 • Huntsville, Alabama 35804 • (205) 533-6987

TABLE OF CONTENTS

FIGURES

TABLES

1. INTRODUCTION

Applied Research, Inc. is pleased to submit this final report on an added scope to Contract NAS8-34337 with NASA/MSFC. This contract has centered on tasks which support the Beta Flight Test Experiment which uses an aircraft-borne laser Doppler velocimeter to measure the aerosol backscatter coefficient. Tasks reported on under this added scope of work are:

A. Using the Beta algorithms developed in previous studies the contractor will examine the 1982 Beta Flight Test Data for determining the Atmospheric Backscatter coefficient.

B. Install a signal processing/enhancement program from earlier studies and provided by the COR on a computer mutually selected by the contractor and the COR.

C. Perform a preliminary analysis to determine the signal processing methods which appear to be prime candidates for combining, to yield enhanced processing capabilities for low signal-to-noise ratios.

D. Demonstrate by simulation the combinations obtained in Task C using the program from Task B.

2. BETA FLIGHT TEST DATA ANALYSIS

Applied Research, Inc. has calculated atmospheric backscatter (β) from data taken by NASA in the summer of 1982. See the Applied Research report to NASA entitled "Beta Experiment Flight Report, Contract No. NAS8-34337, August 1982" for a description of the various flights. This data was taken by a Laser Doppler Velocimeter (LDV) operating in a single particle detection mode. The single particle detection mode utilizes a focus position where the majority of backscatter detections is singular. A focus of 10 meters was determined to be sufficient to encounter single particles the majority of the time.

Applied Research's plan to generate atmospheric backscatter values for the various flights was to produce a computer code incorporating a previously developed algorithm which required the single particle count data (histograms) as input. The output of this computer code is the atmospheric backscatter coefficient calculated from single particle data and is called "single particle β." NASA also requested that Applied Research process the volume data channel. Therefore, code was generated to also output a volume channel β value. A description of the final computer code will be discussed in the following sections and Appendix A contains a complete listing. In order to set up the computer code for automatic processing, several computational quantities had to be defined. These quantities were:

1) Establish correlation between signal value and channel number on the signal processor.

2) Establish the $\frac{S/N}{\sigma}$ value that corresponds to "zero Area", i.e., determine absolute scale for $\frac{S/N}{\sigma}$ vs. Area.

3) L_{eff} at 10 meters.

4) Waist S/N return at 10 meters.

Applied Research had previously performed the LDV system calibration under Contract No. NAS8-34337. However, operation of the LDV at a focus of 10 meters required that additional data be taken to completely define the LDV system. The additional data was taken by W. Jones of NASA and consisted of the data required to calculate L_{eff} at 10 meters and the waist S/N at a 10 meter focus. Applied Research's "Final Report - Beta Experiment" has a description of L_{eff} and how it is calculated for an LDV system.

Signal Processor Channel Calibration data is required to establish the correlation between signal value and signal bin (channel) on the signal processor. This relationship was established by W. Jones of NASA and is shown in Figure 1. This curve was used for all single particle β predictions in this report.

In Applied Research Final Report - Beta Experiment, a plot showing the relative relationship between Signal/Noise/Cross-section ($\frac{S/N}{\sigma}$) and Area (Figure 2) was presented. Since plot has an arbitrary scale, it was necessary to establish an absolute scale for Signal/Noise/Cross-section vs. Area. This is the "single particle calibration" which was done using a sandpaper disk at a focus of 10 meters. The following section contains the theoretical argument for the single particle calibration.

- Single Particle Calibration

In terms of the backscattering elements σ_i of a rotating disk, the signal is $S = \Sigma\, \sigma_i\, g_i$ where g_i is the gain in the region of the i^{th} element. If elements of same value g_i are grouped together into regions denoted by area

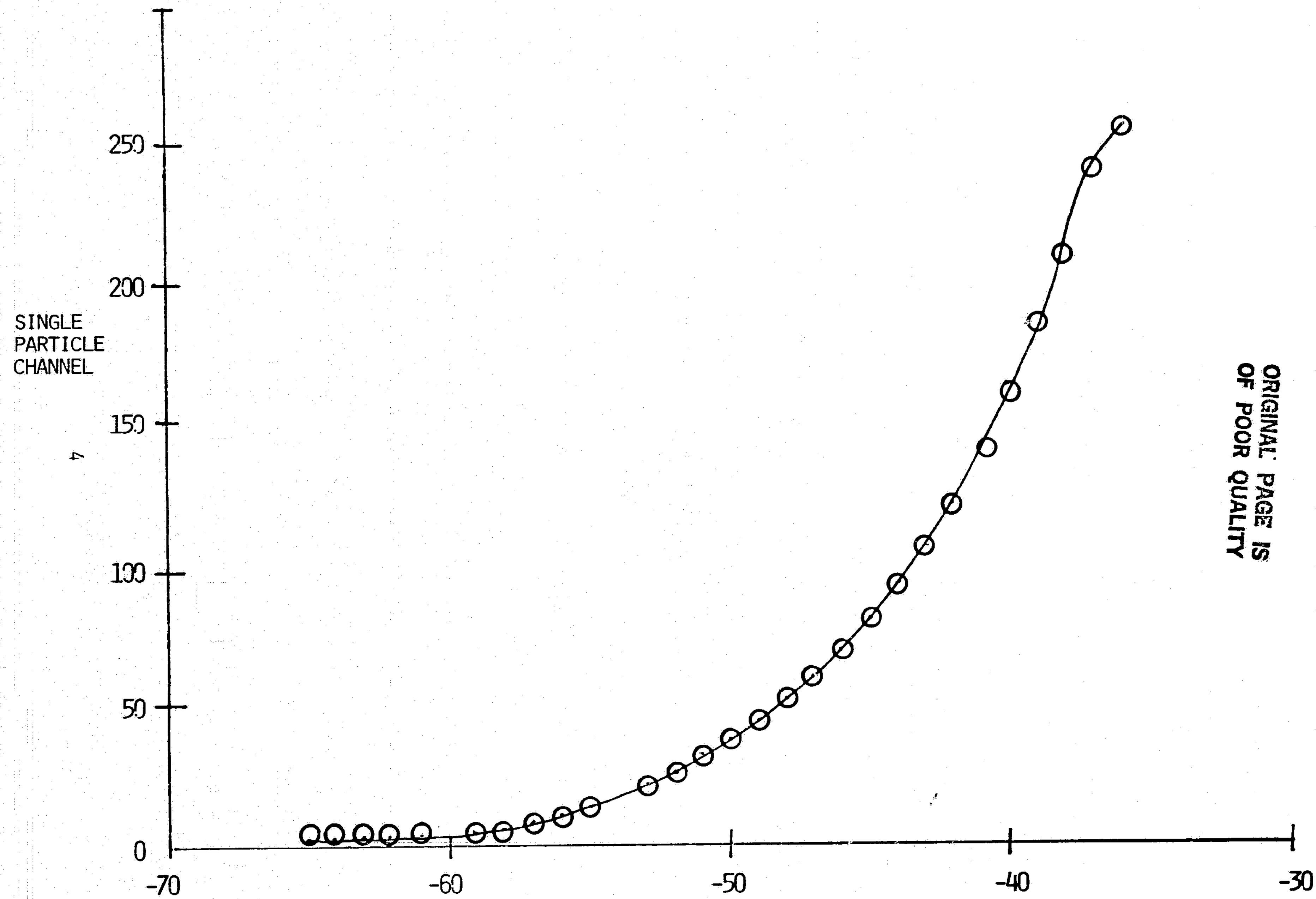


FIG. 1 SIGNAL PROCESSOR CHANNEL VS. SIGNAL LEVEL

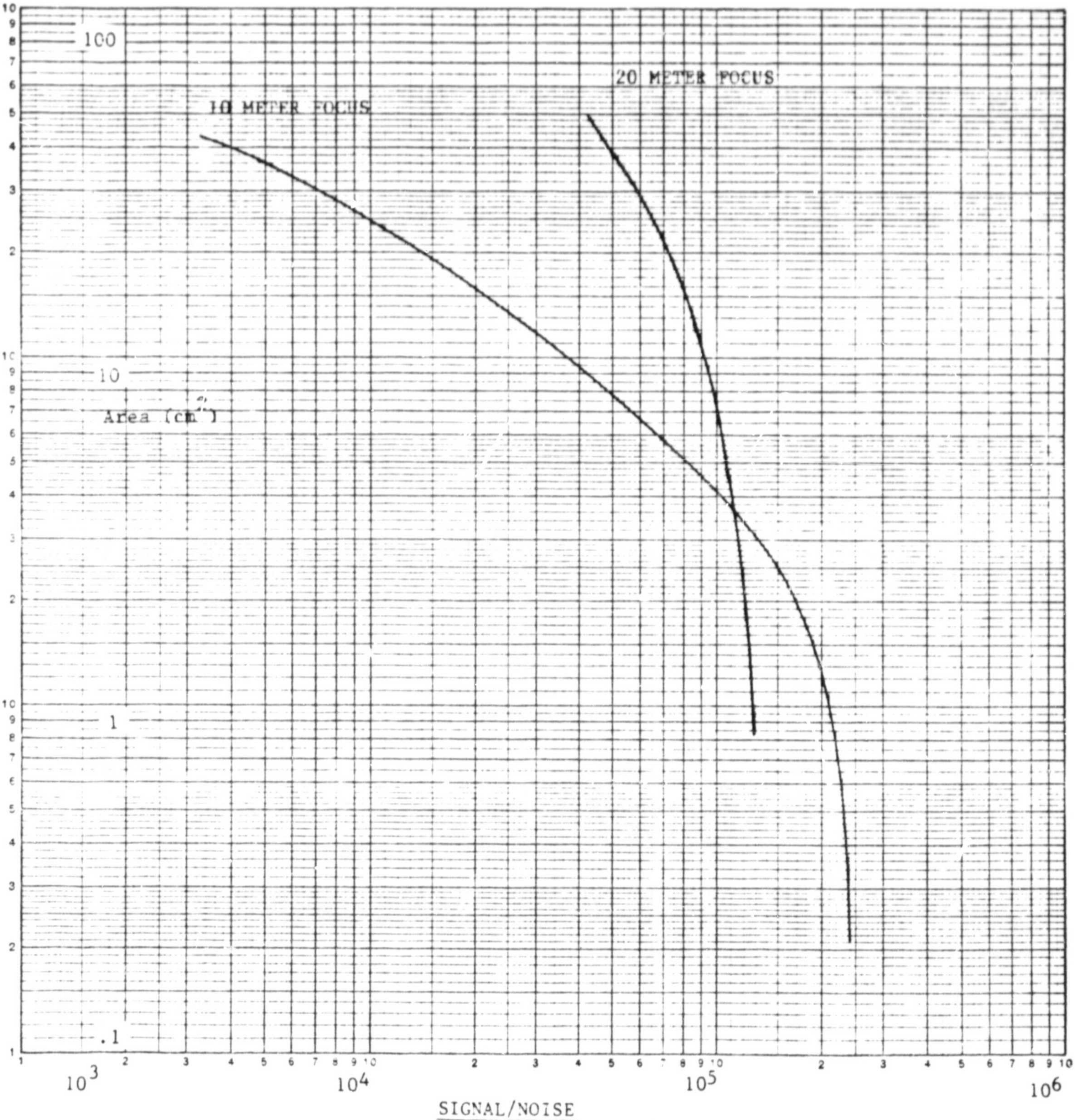


FIGURE 2. S/σ VERSUS AREA (RELATIVE SCALE)

elements "a", then $S = \sum_{a} \left[\sum_{(i \text{ in } a)} \sigma_i \right] g(a)$. Now if $\Sigma\sigma_i = d\sigma$ is a differential, then $S = \sum_{a} \frac{d\sigma}{da} g \, da$. This can be rewritten as $S = \rho\int g(a)da$, where $\rho = \frac{d\sigma}{da}$ is the bidirectional reflectance of the surface and is assumed independent of a. Since $g = S/\sigma$, Figure 2 can be used to obtain the calibration constant C where $g = C \cdot S/\sigma = Cg'$. C is some constant which will give the proper value to the equation $\frac{S'}{\rho} = \int Cg'da$, where S' is the signal return from a sandpaper disk at 10 meters, and g' is the relatively scaled S/N/σ data. Figure 3 shows the calibration data which resulted from this exercise. This data allows a signal/noise/sigma to be associated with a particular area at a 10 meter focus.

By using the equation $\frac{S}{\sigma}\Big|_{o} = g(o)$ to tie-down the Area curve (Figure 3) at an Area of 0, a cross-section can be produced for each signal bin on the signal processor. This association of cross-section with signal bin was calculated for each single particle β prediction.

Algorithm Description

When all of the required information was completed, Applied Research proceeded to set up a computer code which would automate the data predictions. The computer prediction code has 3 major sections which are repeated over and over until the data is exhausted. The first section is a subroutine (GETDAT) to read the data off disk and translate the proper values to work arrays. The second section does the noise calculations for both the volume and single particle channels of the data. Both calculations are done by using the data to "look up" the noise values in a table. A linear interpolation is done in the table to obtain the final values. The third section is the subroutine (DATANAL) which is the single particle β inversion routine.

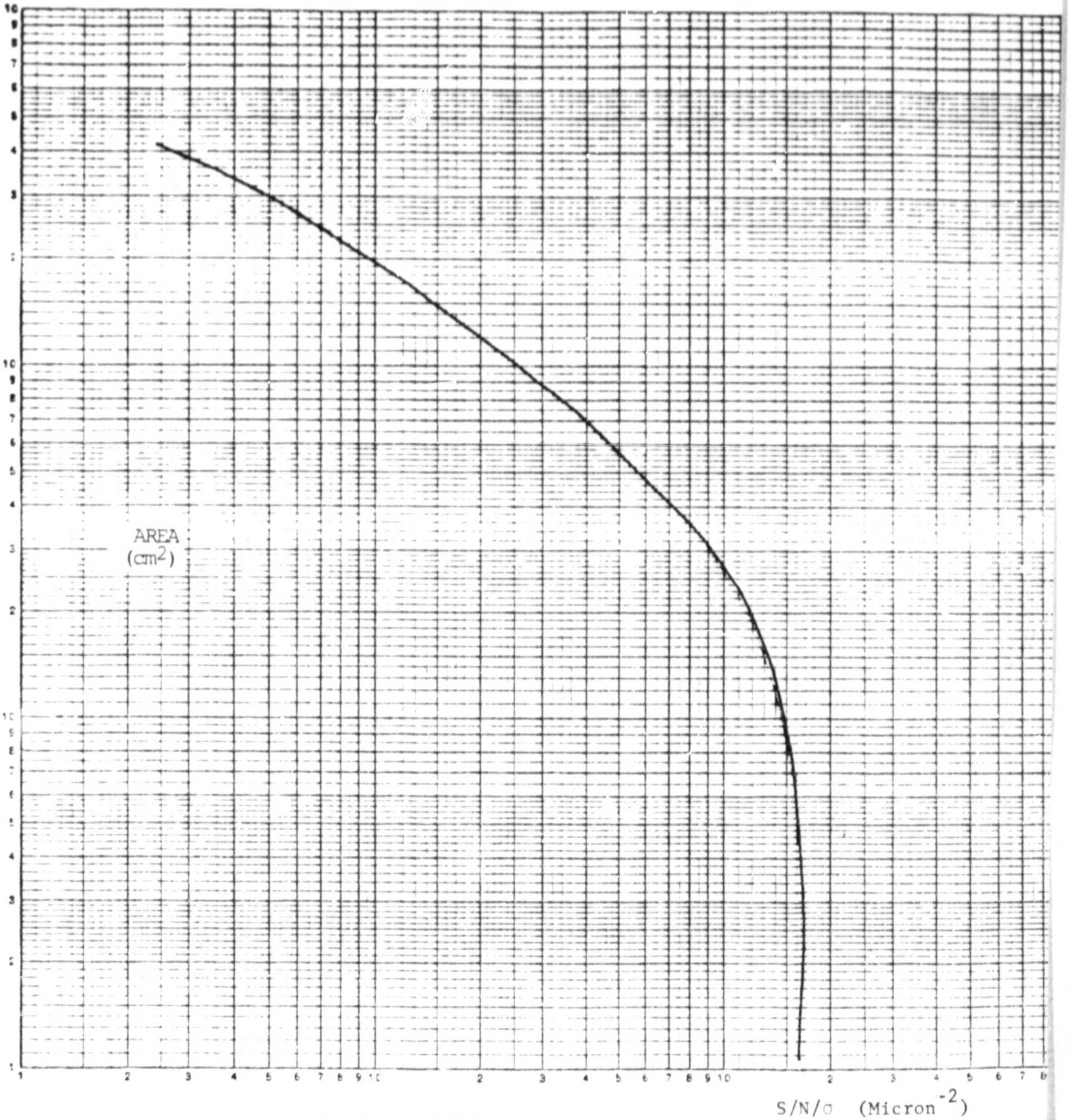


FIG. 3 S/N/σ VERSUS AREA
(Absolute Scale)

The BETA Data is packed into 768 byte records of the Signal V computer. Each record of data must be decoded into the 384 words which are identified in Table 1. The alternating noise samples were used to calculate a "noise factor" for each set of data processed by the algorithm. This was done by monitoring the integrated volume β word in the noise set of data and then calculating what noise (in dbm) corresponded to this value by using a "calibration table look up" with linear interpolation. The noise data blocks were further utilized by subtracting off the noise hits in each bin for the single particle BETA Data. Care was taken to compensate for bandwidth differences between single particle noise and volume channel noise data.

The Beta Inversion Algorithm requires approximately 10,000 particle hits to be accurate; therefore, the data had to be combined in most cases in order to obtain the 10,000 particle count minimum. When this was done, the time that corresponds to the data was also averaged over the data-taking interval. Therefore, the output of the algorithm was single particle β versus time. However, there is not a one-to-one correspondence between input data records and output data points due to the combining which occurs when the number of single particle hits is low.

A volume backscatter coefficient was also calculated for the same data as for the single particle β. This data requires integration since the LDV was focused at 10 meters where it is likely that most of the time single particles occupied the sensitive focal volume of the LDV. A volume β is normally defined as the backscatter return from a collection of particles measured at an instant. The volume Beta was calculated by using the following equation:

$$\beta = \frac{V_s/V_n}{S^\circ_d} \quad \frac{\rho_o}{L_{eff}}$$

where V_s is the integrated volume single, V_n is the integrated volume noise,

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

Table 1. DATA RECORD

Ø-	TIMH:	12	CURRENT TIME
	TIMM:	Ø	
	TIMS:	Ø	
	TIMF:	10°°	
4-	TIMH1:	12	TIME AT START OF RECORD
	TIMM1:	Ø	
	TIMS1:	Ø	
	TIMF1:	Ø	
8-	DATEM:	6	CURRENT DATE
	DATED:	1	
	DATEY:	32	
11-	STEPR1:	Ø	STEPPER POSITIONS (NOT PRESENTLY USED)
	STEPR2:	Ø	
	STEPR3:	Ø	
14-	VOLBH:	Ø	VOLUME BETA INTEGRATORS
	VOLBL:	Ø	
	VOLBN:	Ø	
19-	PARMS:	171356	CONTROL REGISTER PARAMETERS
	CLOCKP:	11.	DIGITAL FILTER CLOCK PERIOD
	FILTER:	2	IF FILTER WIDTH
20-	TIMEON:	3	VIDEO TIME CONSTANT
	VCOFRQ:	140.	VCO FREQUENCY
	DOFSET:	Ø 11.	A/D FILTER BIAS
23-	DISCRM:	Ø	DISCRIMINATOR THRESHOLD
	IFGAIN:	45.	IF GAIN
25-	FWDID:	Ø	FORWARD/AFT IDENTIFIER
	NDSETS:	1	# SCANS PER WRITE
	INTSEC:	1ØØØ.	MS PER INTEGRATION PERIOD
	ADDASV:	Ø	UP IF ADDAS LINK ON
	ADDASI:	1ØØ.	# MS PER ADDAS OUTPUT
30-	SCAN:	Ø	UP IF SCANNER ENABLED
	TASADD:	3ØØ	TAS FROM ADDAS
	TASKB:	3ØØ	TAS FROM KEYBOARD
	THETA:	5ØØ.	ANGLE THETA IN .Ø1-DEG UNITS
34-	VCOOFS:	Ø	VCO OFFSET IN 53-KHZ UNITS
88	IDENT:	BLHB 8Ø.	IDENTIFICATION STRING
128			
383	DATA		

S_d^* is the signal to noise return of a sandpaper disk at 10 meters, ρ_o is the bidirectional reflectance of sandpaper and L_{eff} is the effective length of the LDV while focused at 10 meters. L_{eff} was calculated by using data generated by W. Jones at 10 meters which is shown in Figure 4. The corresponding L_{eff} is plotted in Figure 5, which shows how this data point fits in with other data points calculated by Applied Research.

The integrated signal, V_s, in the volume channel was calculated by subtracting off the average noise signal (V_n) for that time interval. The signal return from a sandpaper disk at 10 meters at the focal volume waist and the bidirectional reflectance of the sandpaper disk were supplied by W. Jones of NASA.

Prediction Results

A total of 23 flights were processed by Applied Research. Some flights had problems with the data which were caused by data transfer, but the large majority did not. The prediction algorithm had only two cases where it did not attempt to process the data: (1) If the airplane airspeed is less than 250 knots (such as at takeoff or landing) and (2) if the last signal bin contained more than 5,000 hits. Case No. 2 occurs whenever dense clouds are being flown through and since the algorithm predictions are inadequate here, the data is ignored. Figures 6 through 27-A show plots of single particle β and volume β as a function of time. Figures 28 through 49 show the average aerosol backscatter cross-section as a function of time. Figures 50 through 71 show the atmospheric aerosol number density as a function of time. The time on each plot is in minutes, calculated by multiplying the military hour by 60 and adding the minutes. Note that Flight 12 is missing completely

ORIGINAL PAGE IS
OF POOR QUALITY

because none of the data was processable by the single particle prediction algorithm. Other gaps in the data occur because of the two cases listed above. Note that there are some single particle β predictions on some flights which get very large. These instances correspond to cloud cases. In general those large values of β cannot be considered as accurate since the algorithm was not set up to handle high density large particles. The corresponding average cross-sections and densities would also be suspect. Figure 27-A shows the result of processing with only 1000 single particle hits. These results are not significantly different from those for 10,000 hits and indicate that shorter flight intervals per output beta point may be possible. Further analysis of this question is required.

The single particle and volume backscatter coefficients differ by a constant offset, the origin of which is not presently understood. This error may be found in the calibration techniques or processor assumptions. An evaluation of the uncertainties in the backscatter coefficient determinations would illuminate this problem.

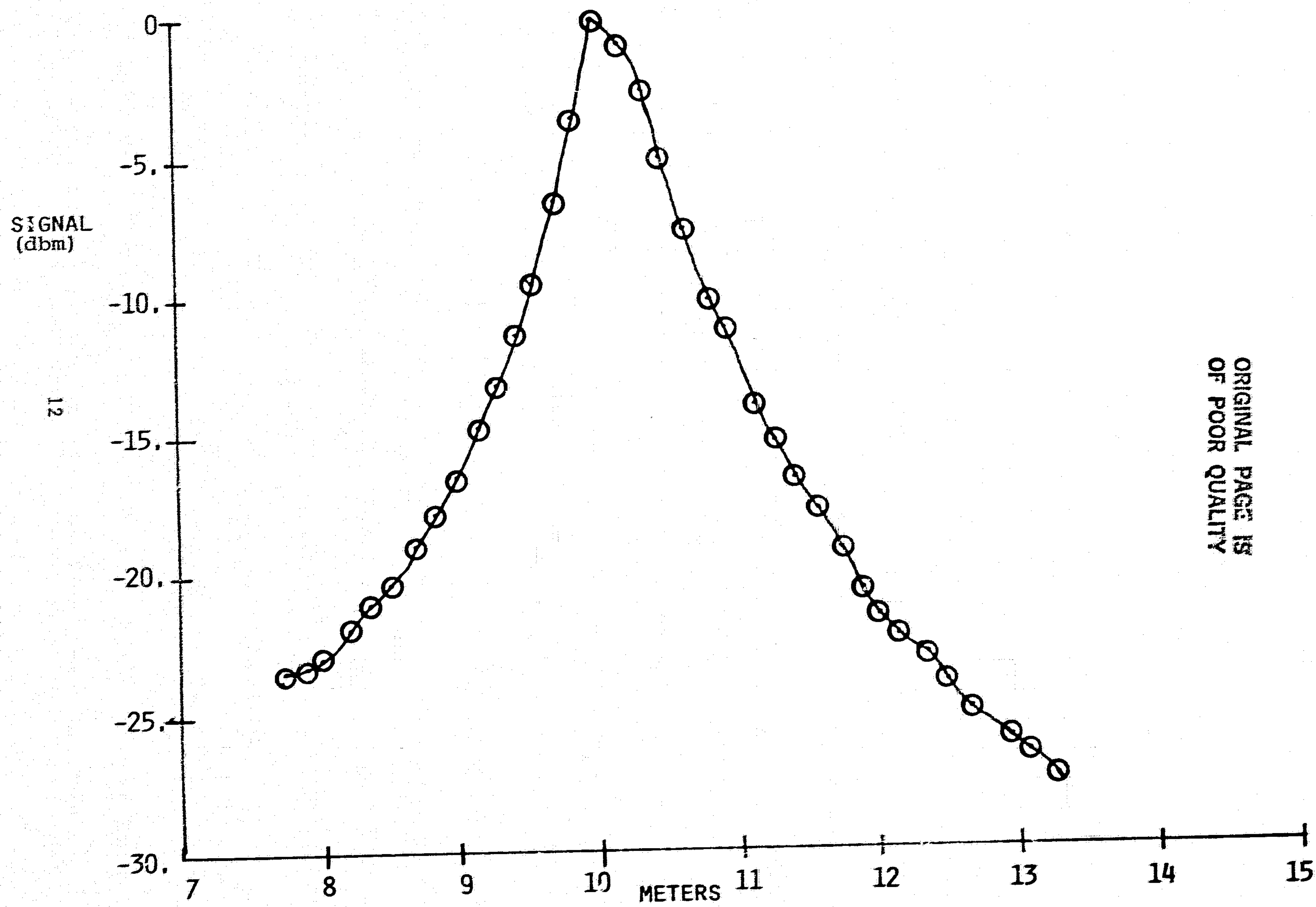


FIG. 4 LONGITUDINAL PROFILE AT 10 METER FOCUS

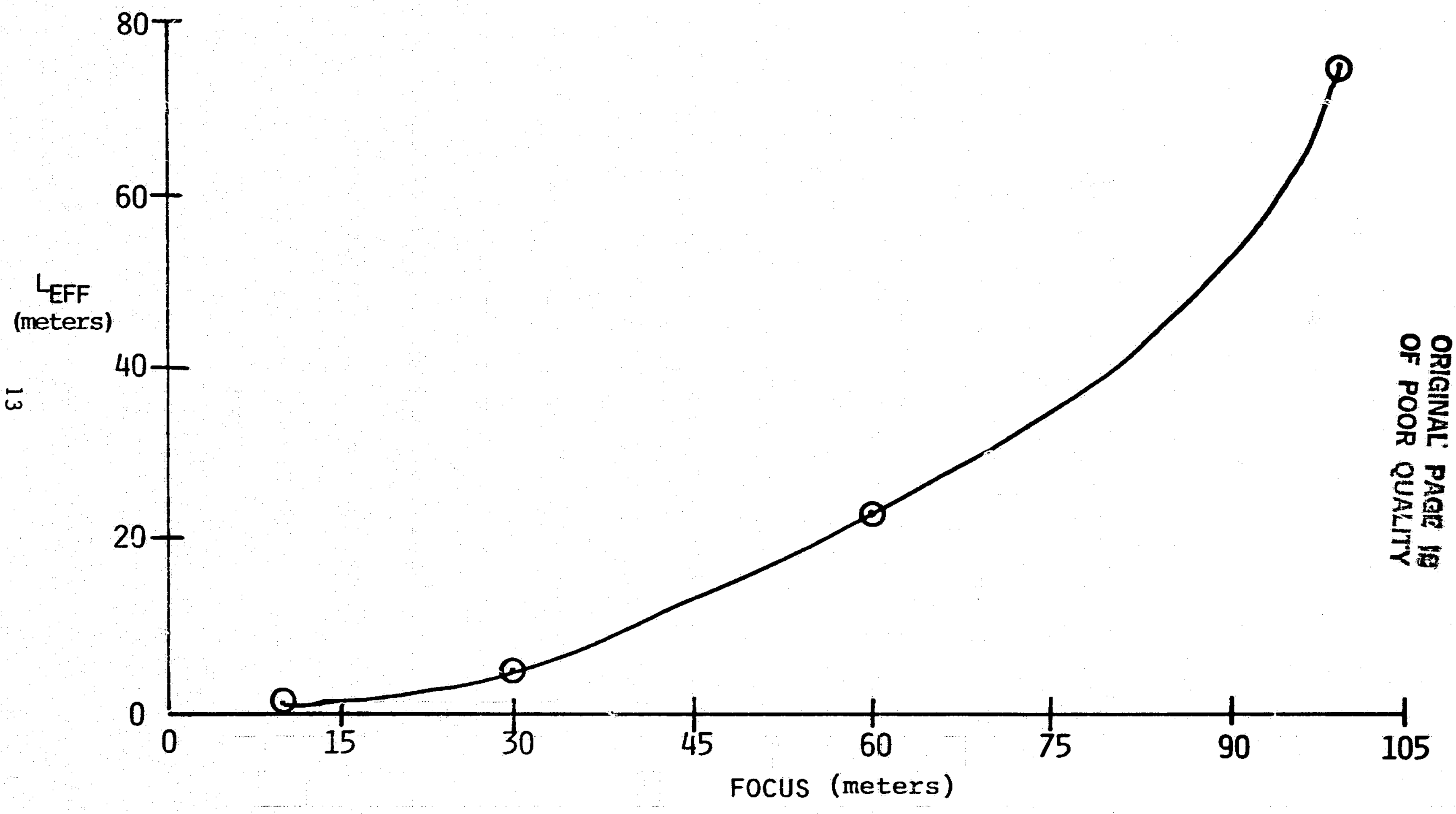


FIG. 5 L_{EFF} vs. FOCUS

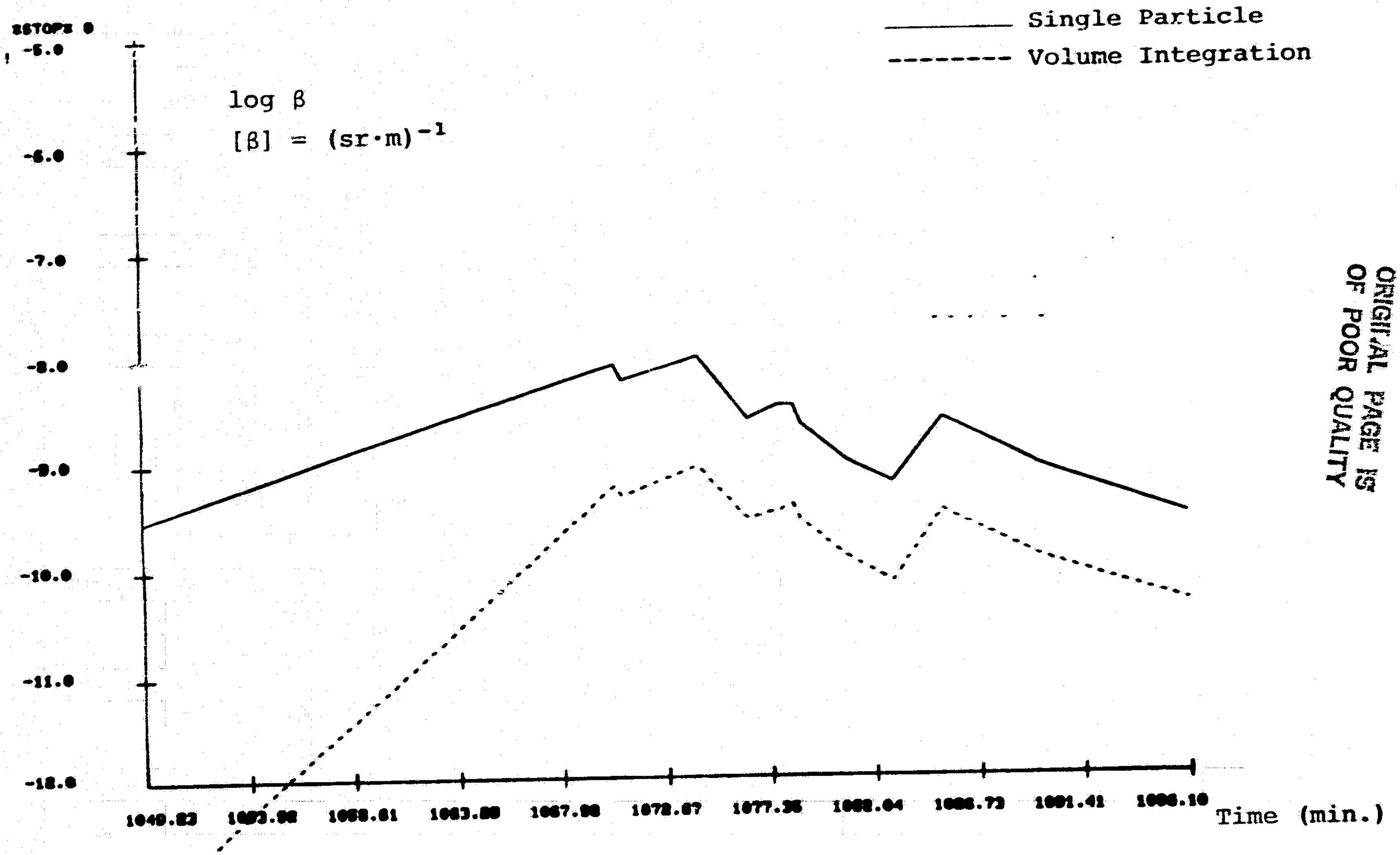


Fig. 6 Backscatter Coefficient β

ORIGINAL PAGE IS
OF POOR QUALITY

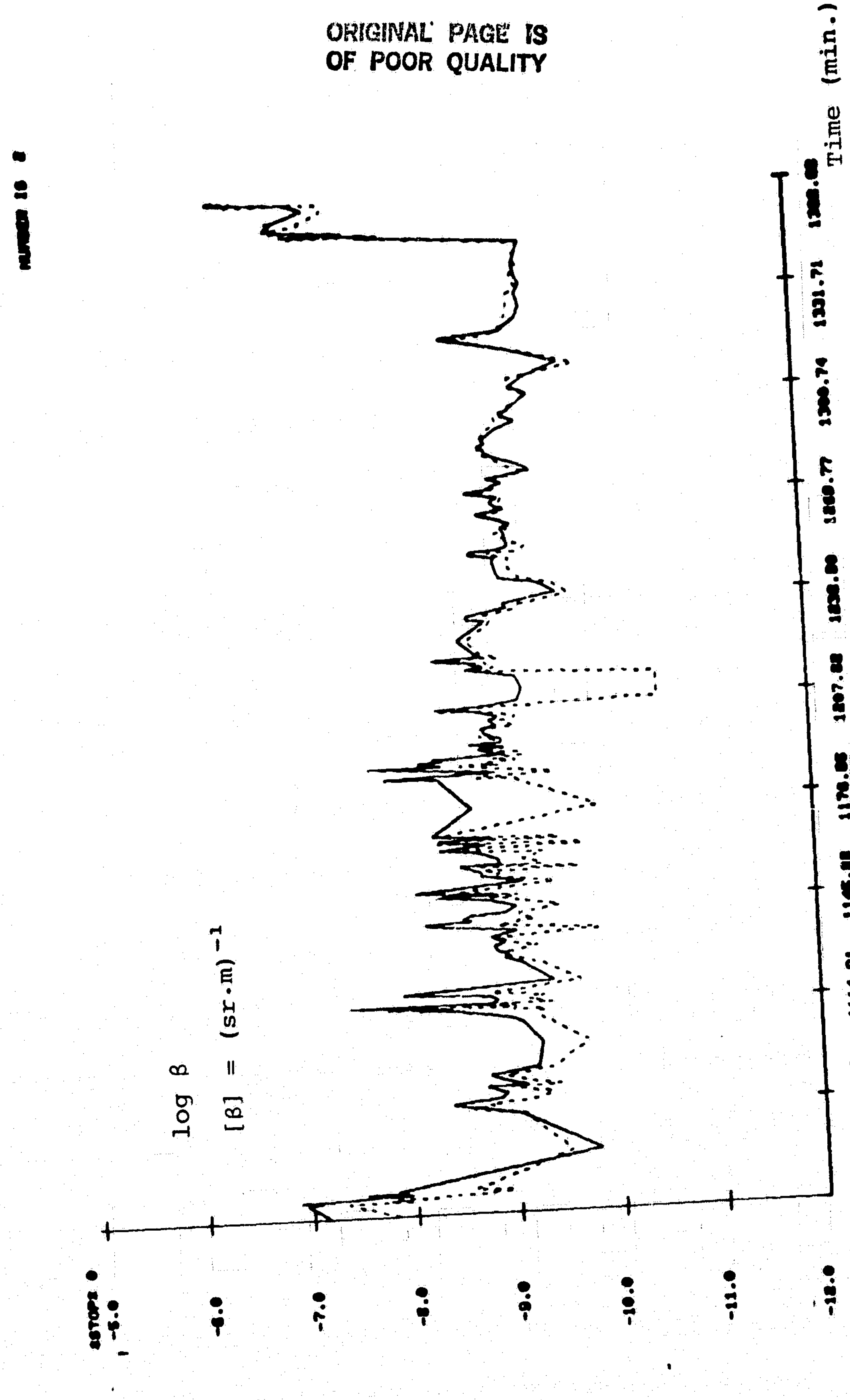


Fig. 7 Backscatter Coefficient β

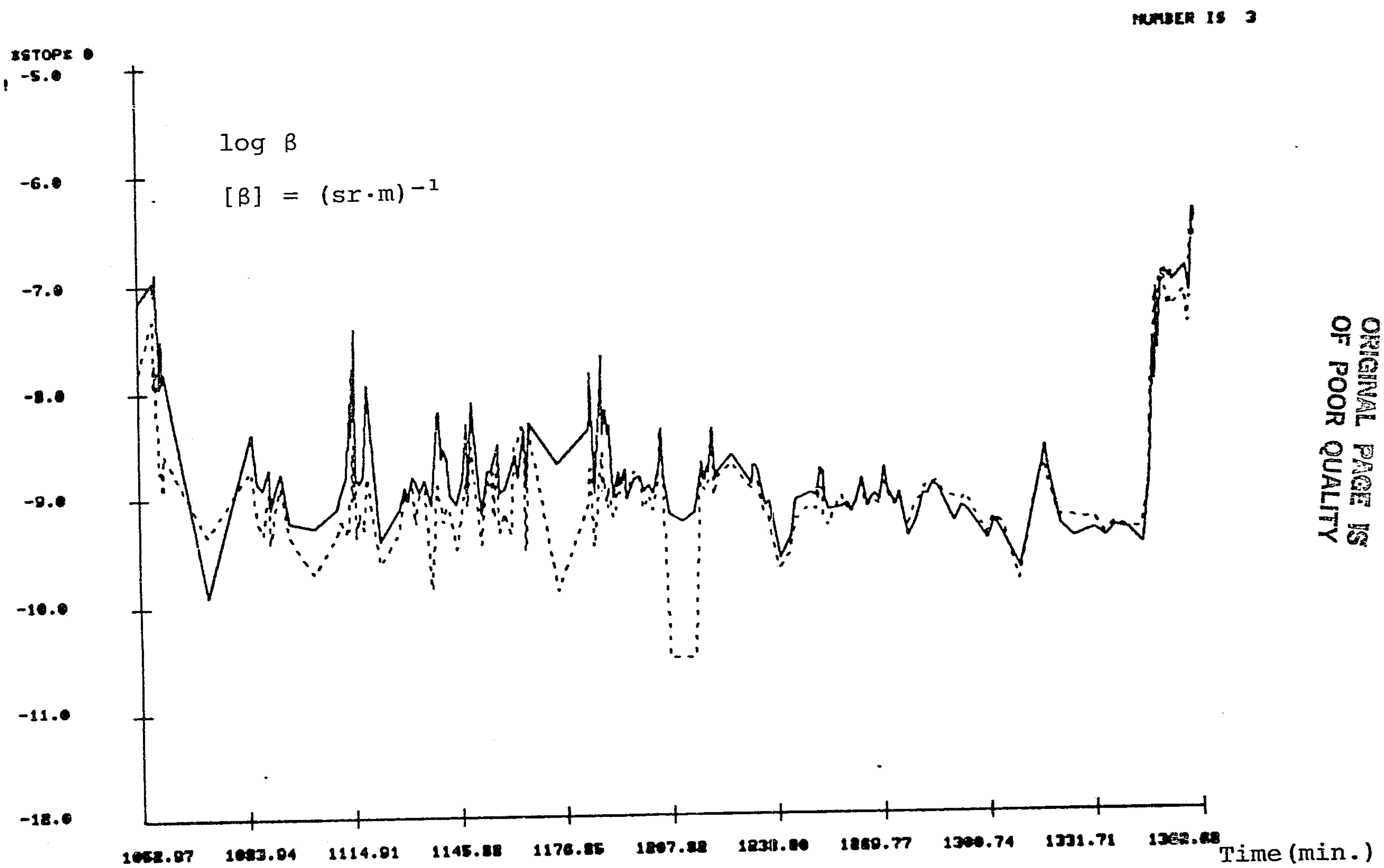


Fig. 8 Backscatter Coefficient β

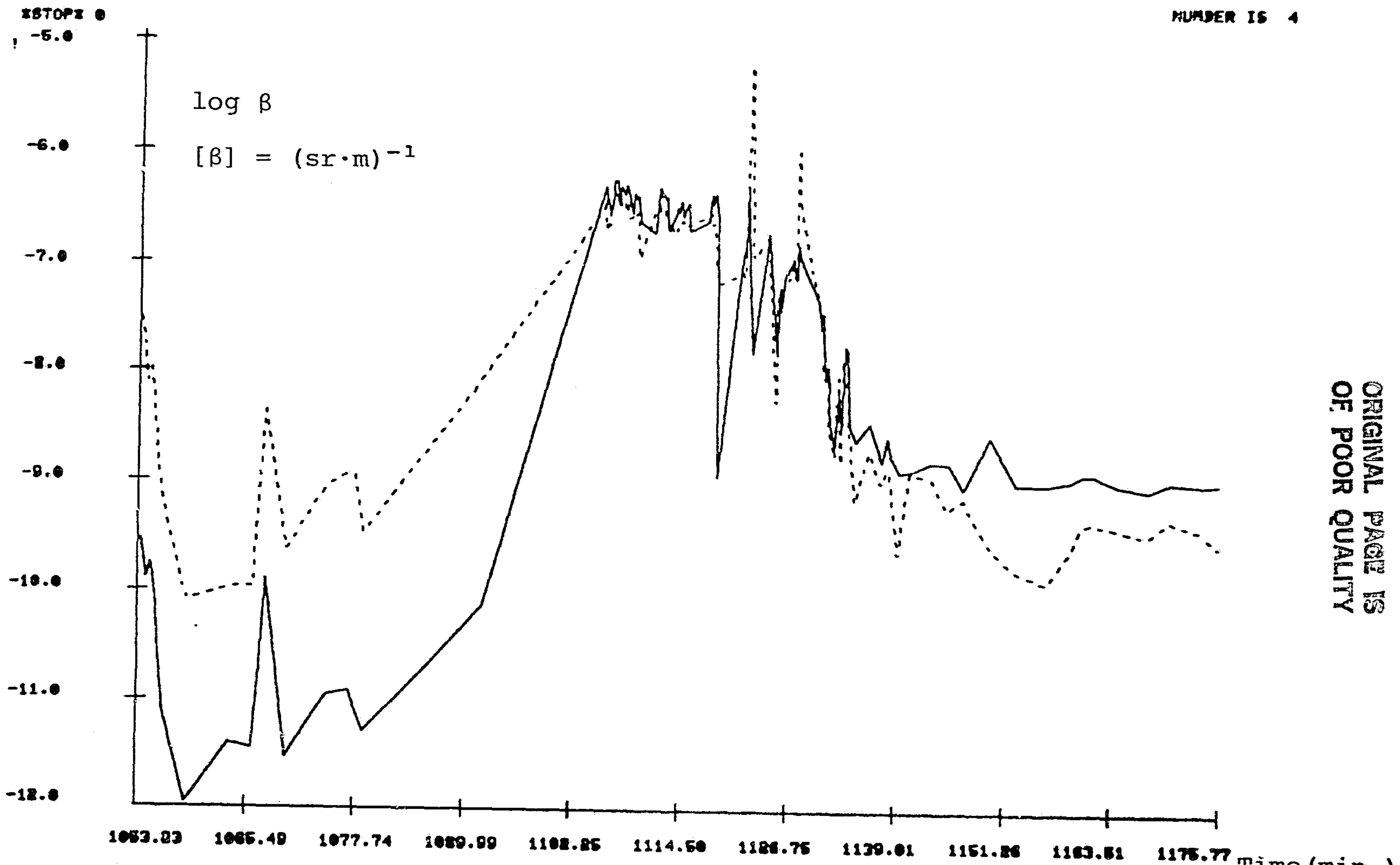


Fig. 9 Backscatter Coefficient β

ORIGINAL PAGE IS
OF POOR QUALITY

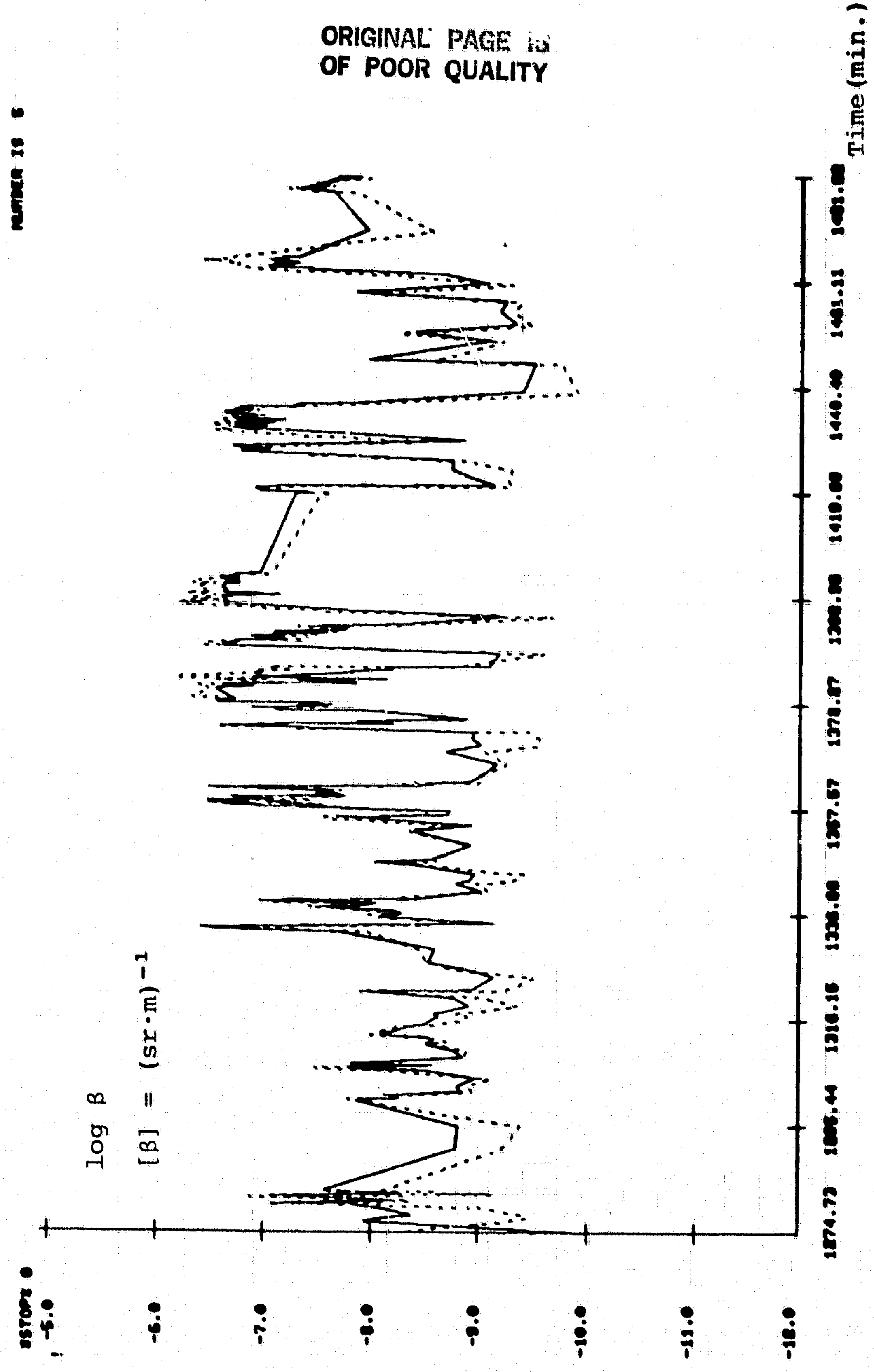


Fig. 10 Backscatter Coefficient β

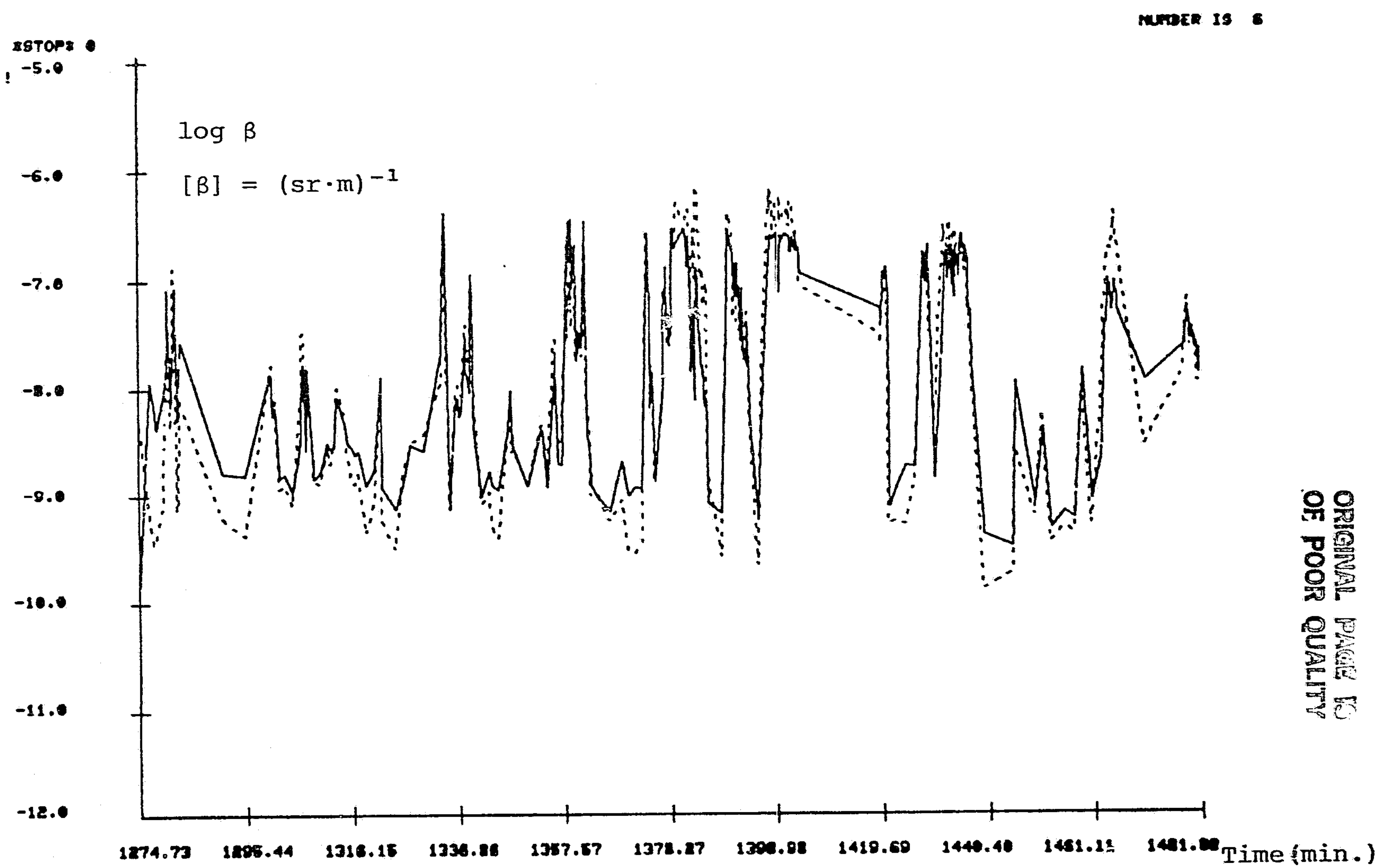


Fig. 11 Backscatter Coefficient β

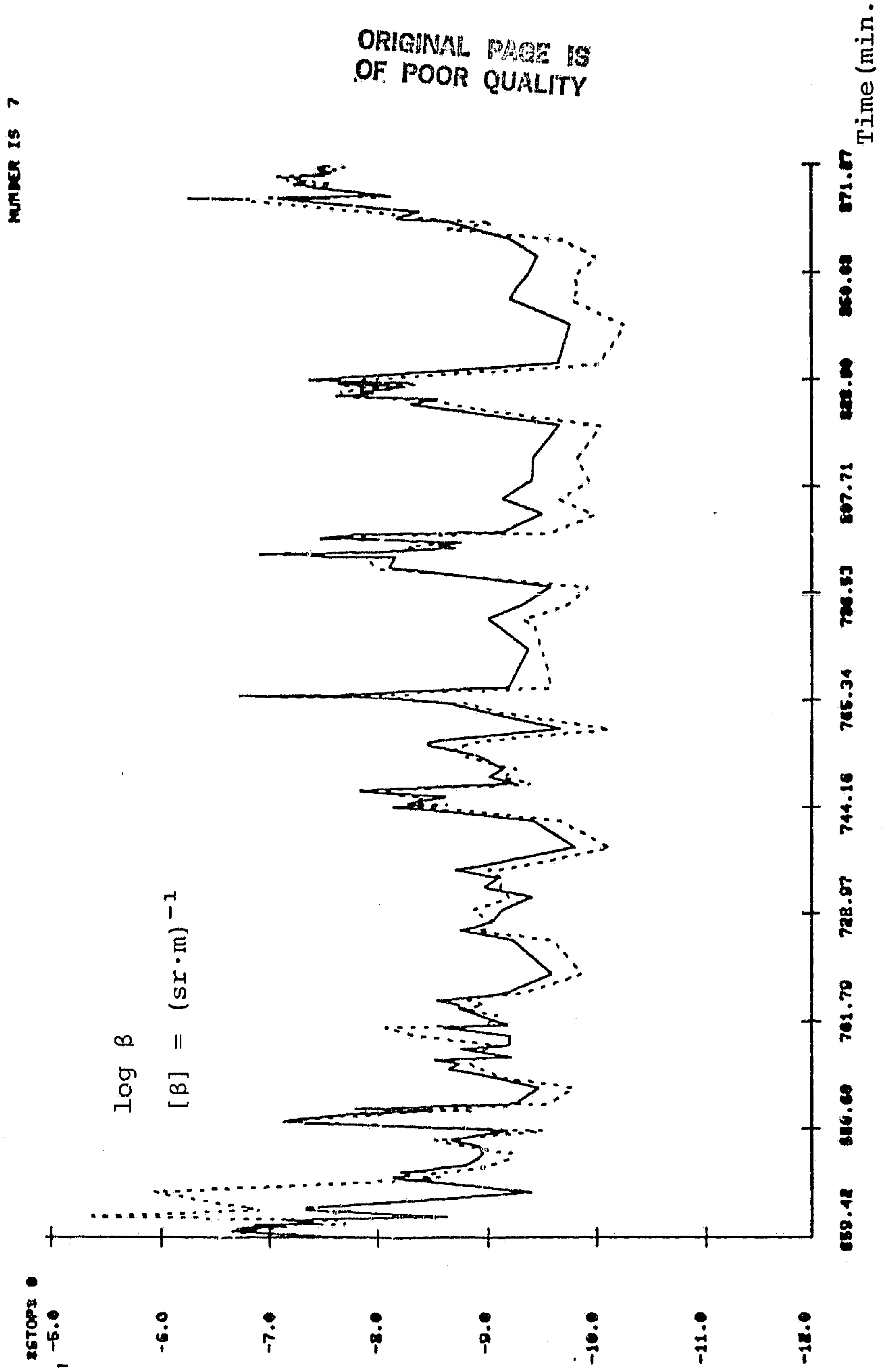


Fig. 12 Backscatter Coefficient β

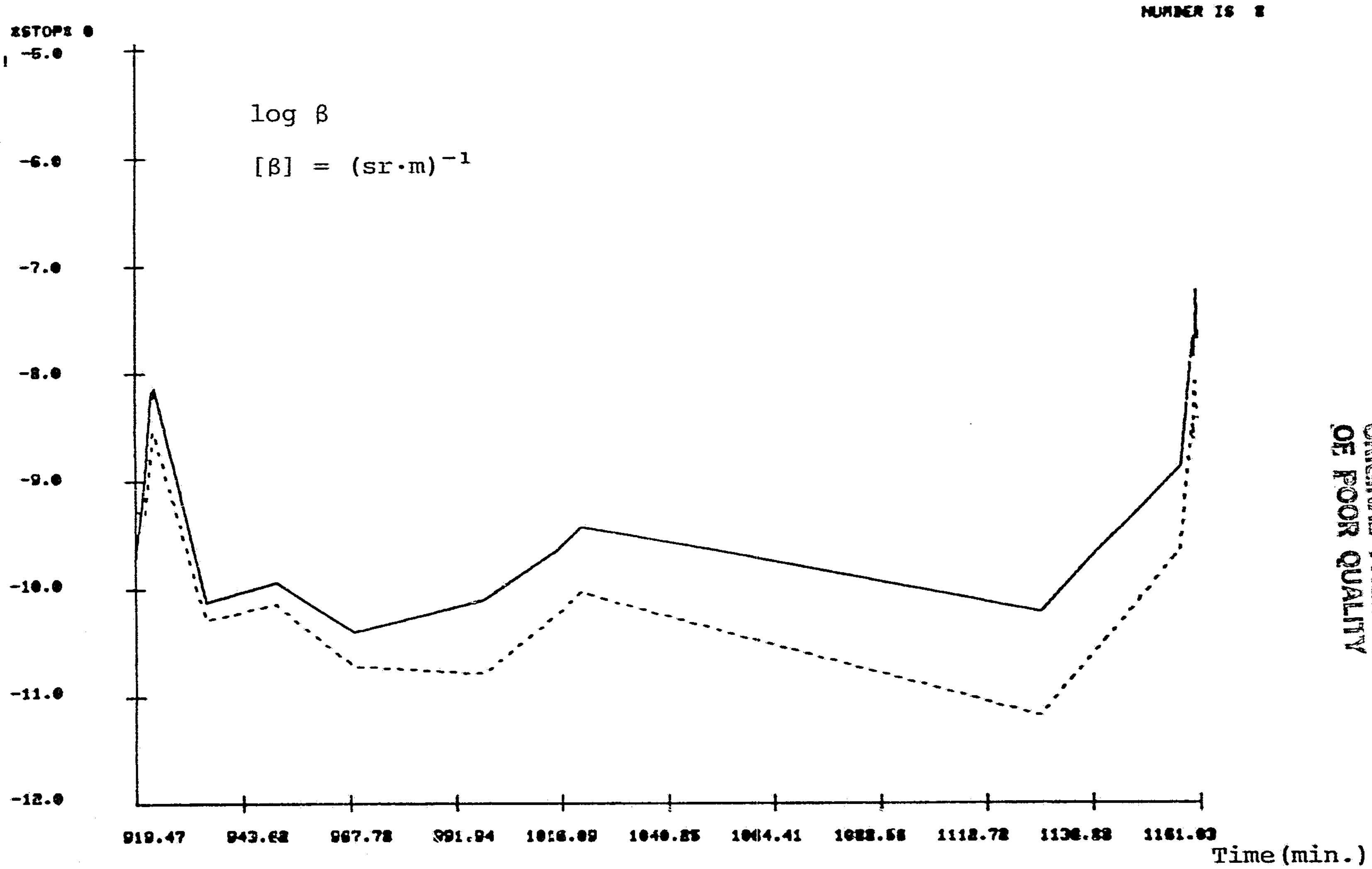


Fig. 13 Backscatter Coefficient β

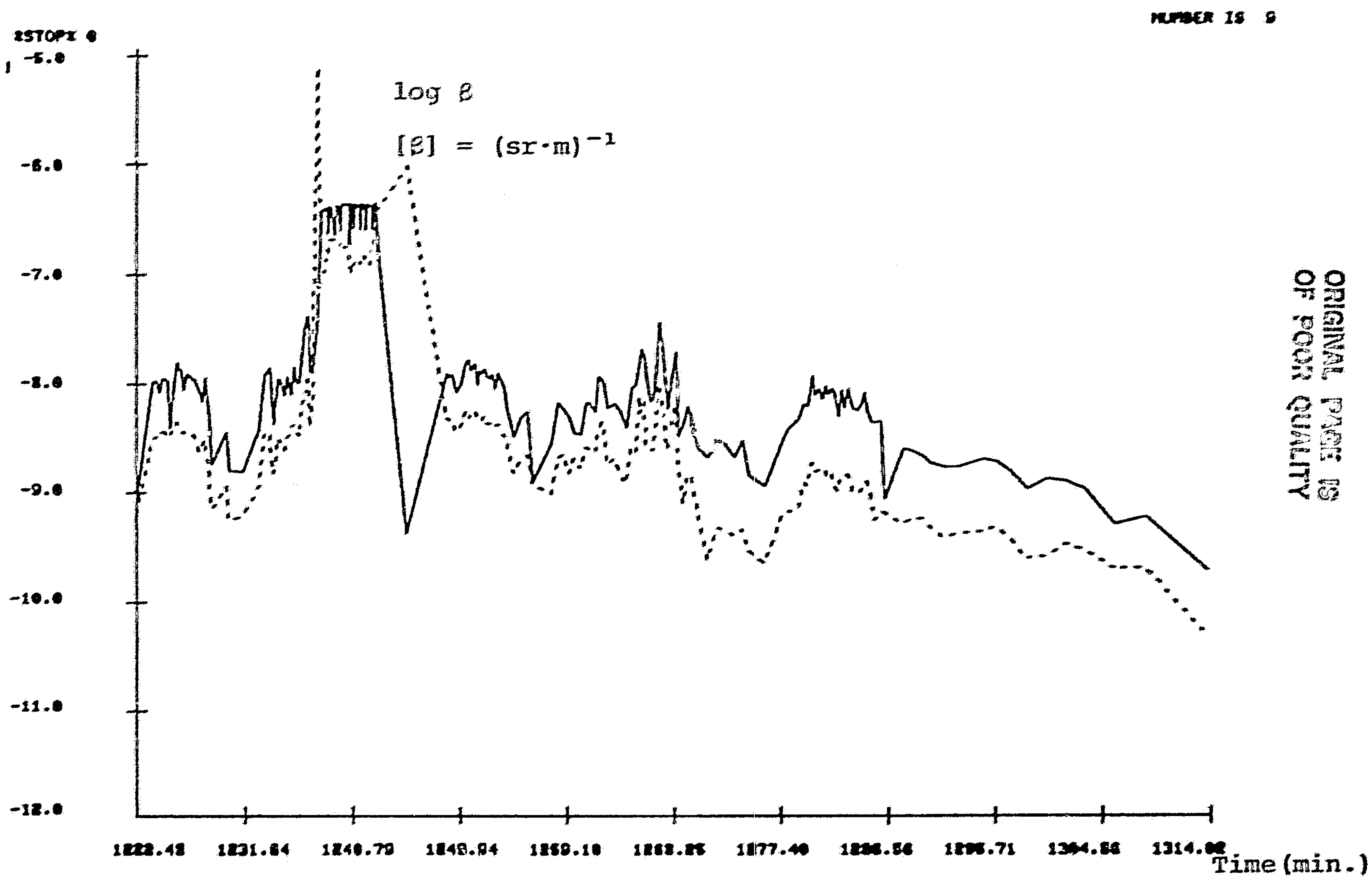


Fig. 14 Backscatter Coefficient β

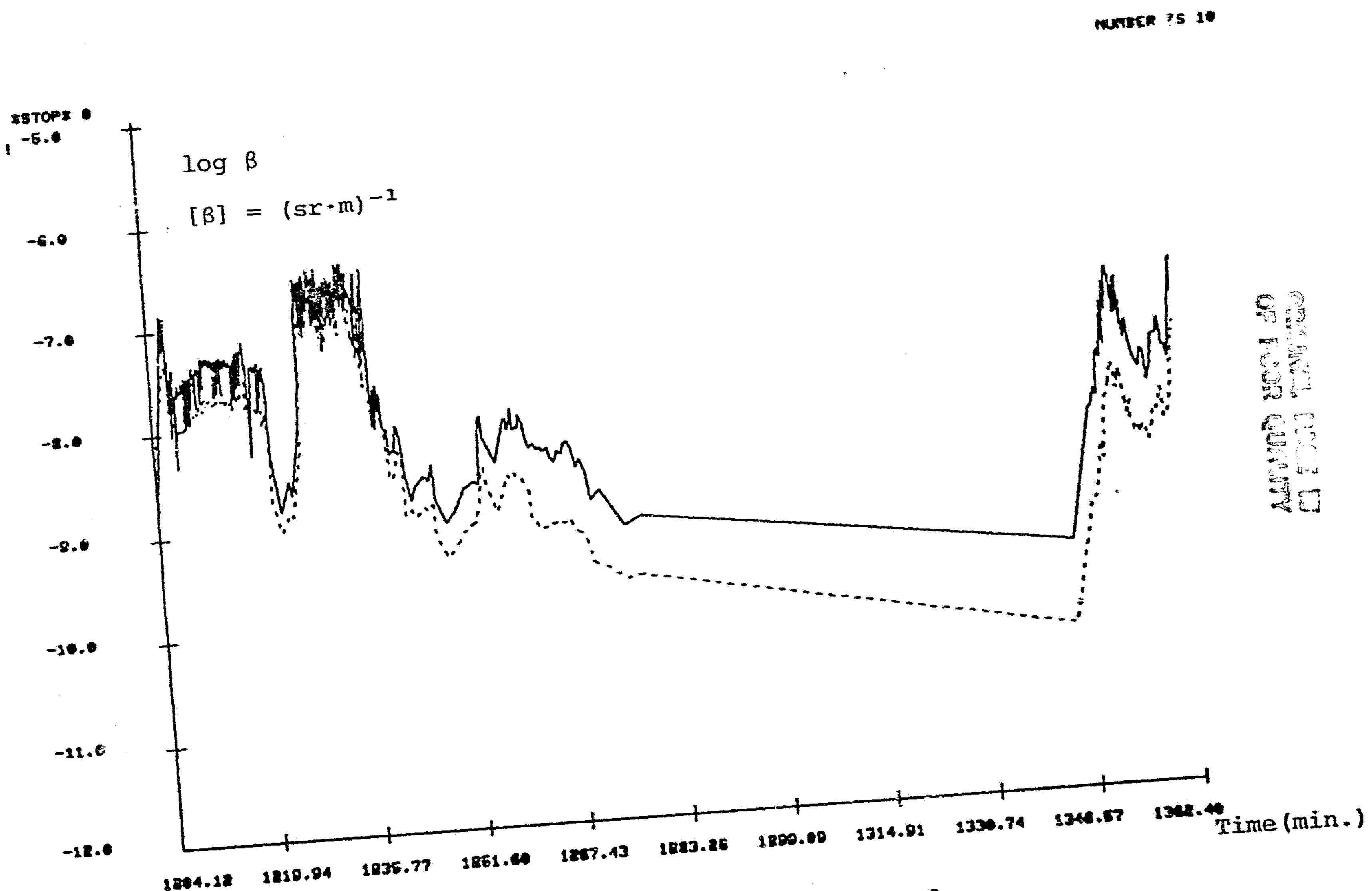


Fig. 15 Backscatter Coefficient β

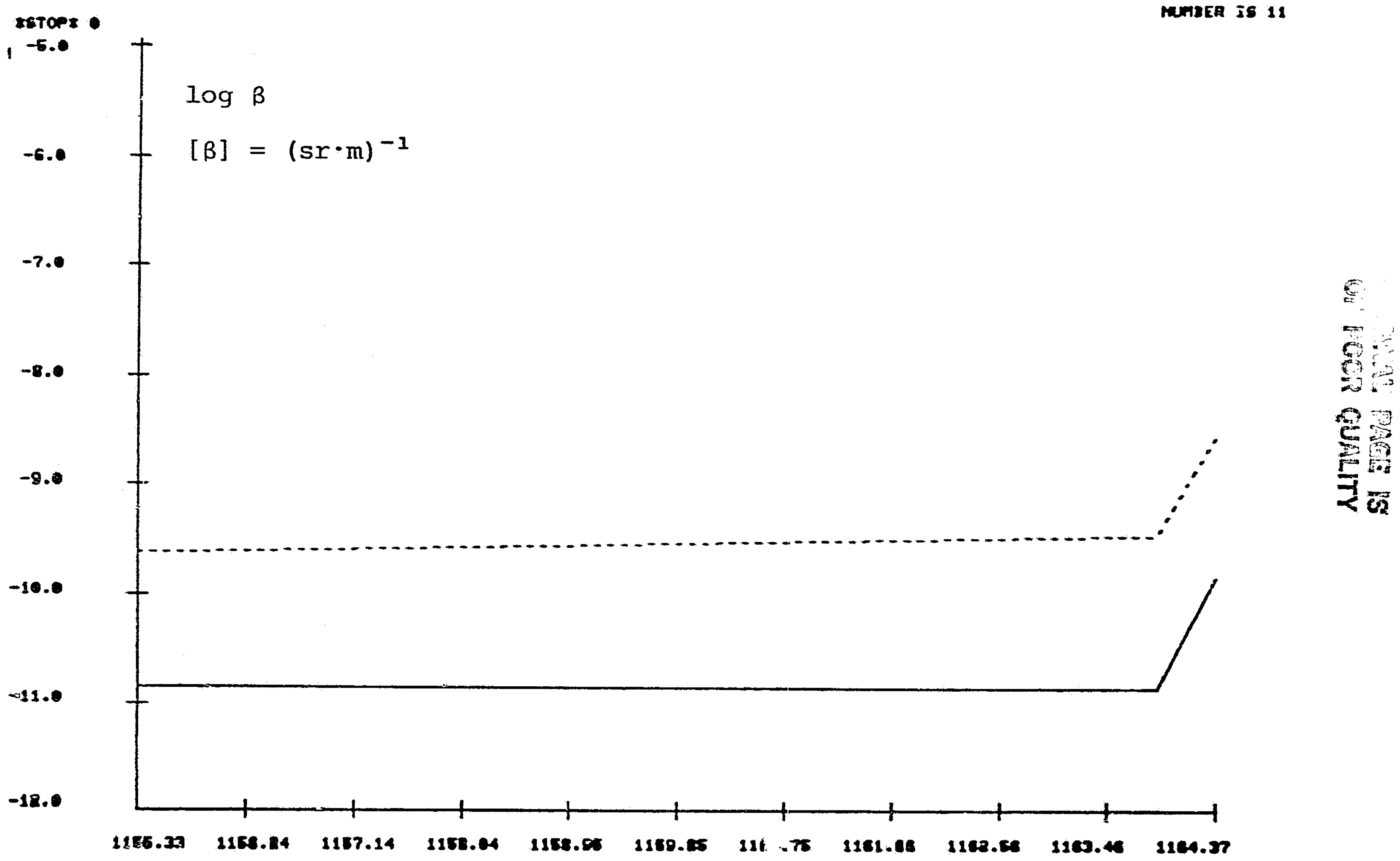


Fig. 16 Backscatter Coefficient β

ORIGINAL PAGE IS OF POOR QUALITY

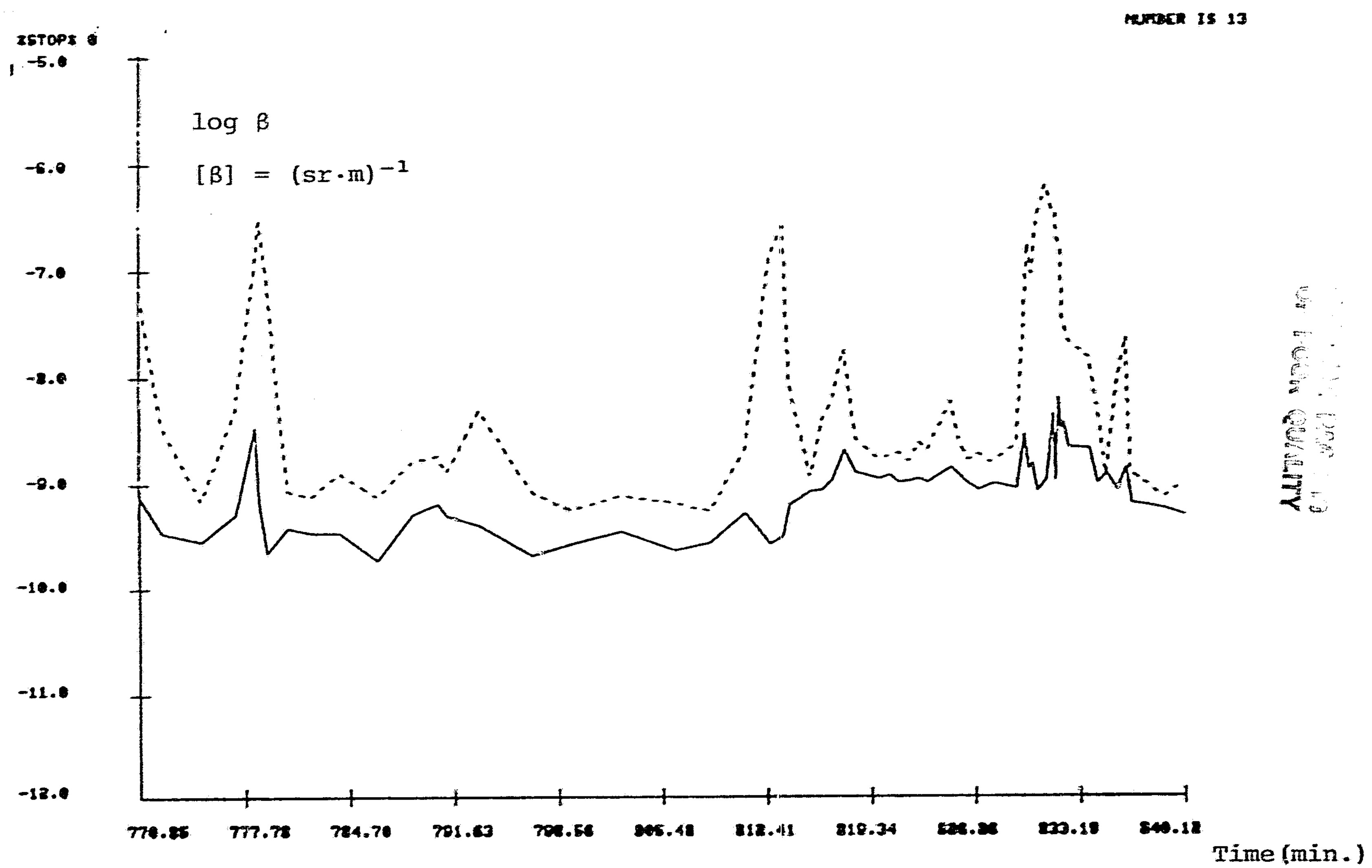


Fig. 17 Backscatter Coefficient β

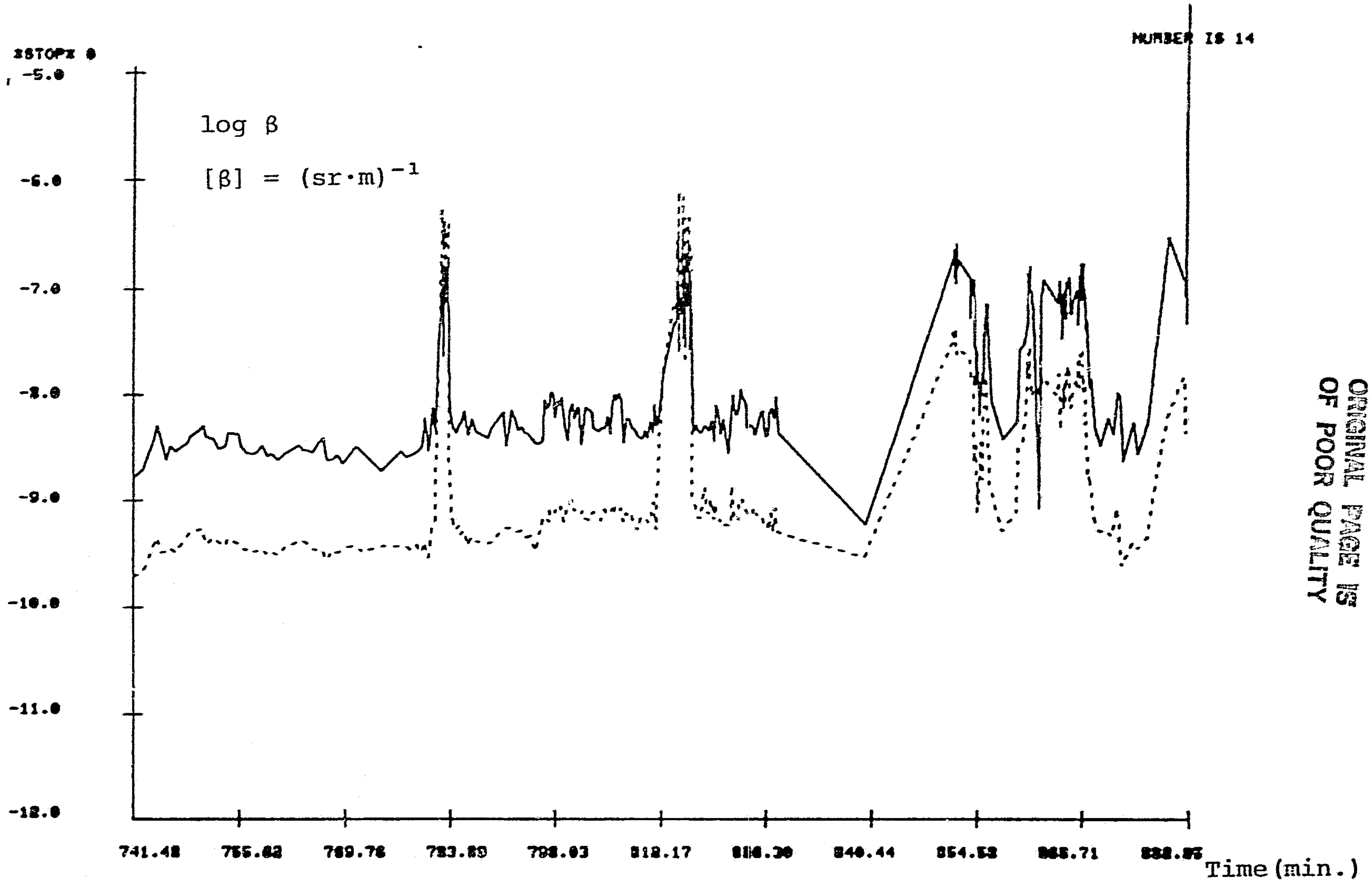


Fig. 18 Backscatter Coefficient β

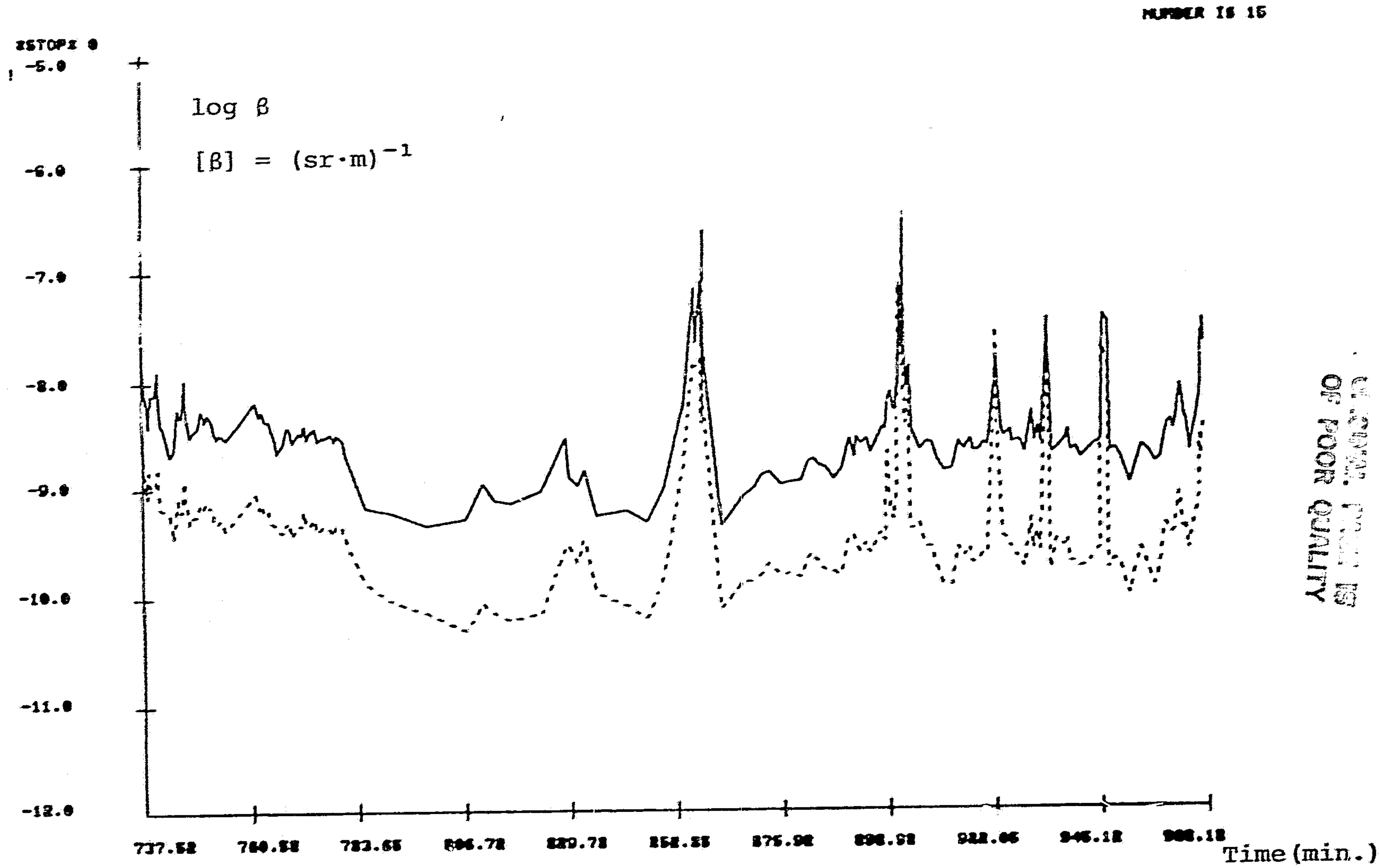


Fig. 19 Backscatter Coefficient β

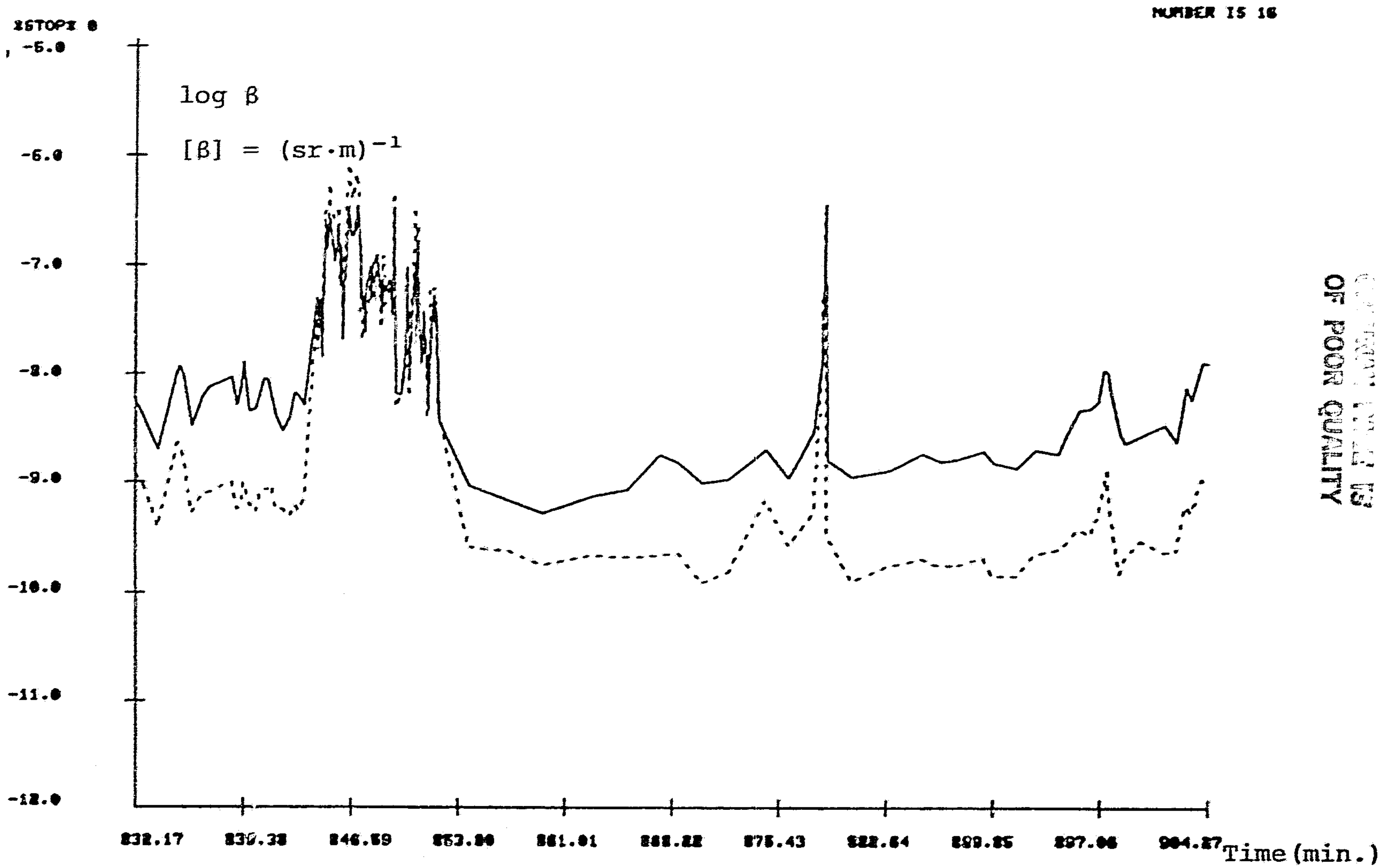


Fig. 20 Backscatter Coefficient β

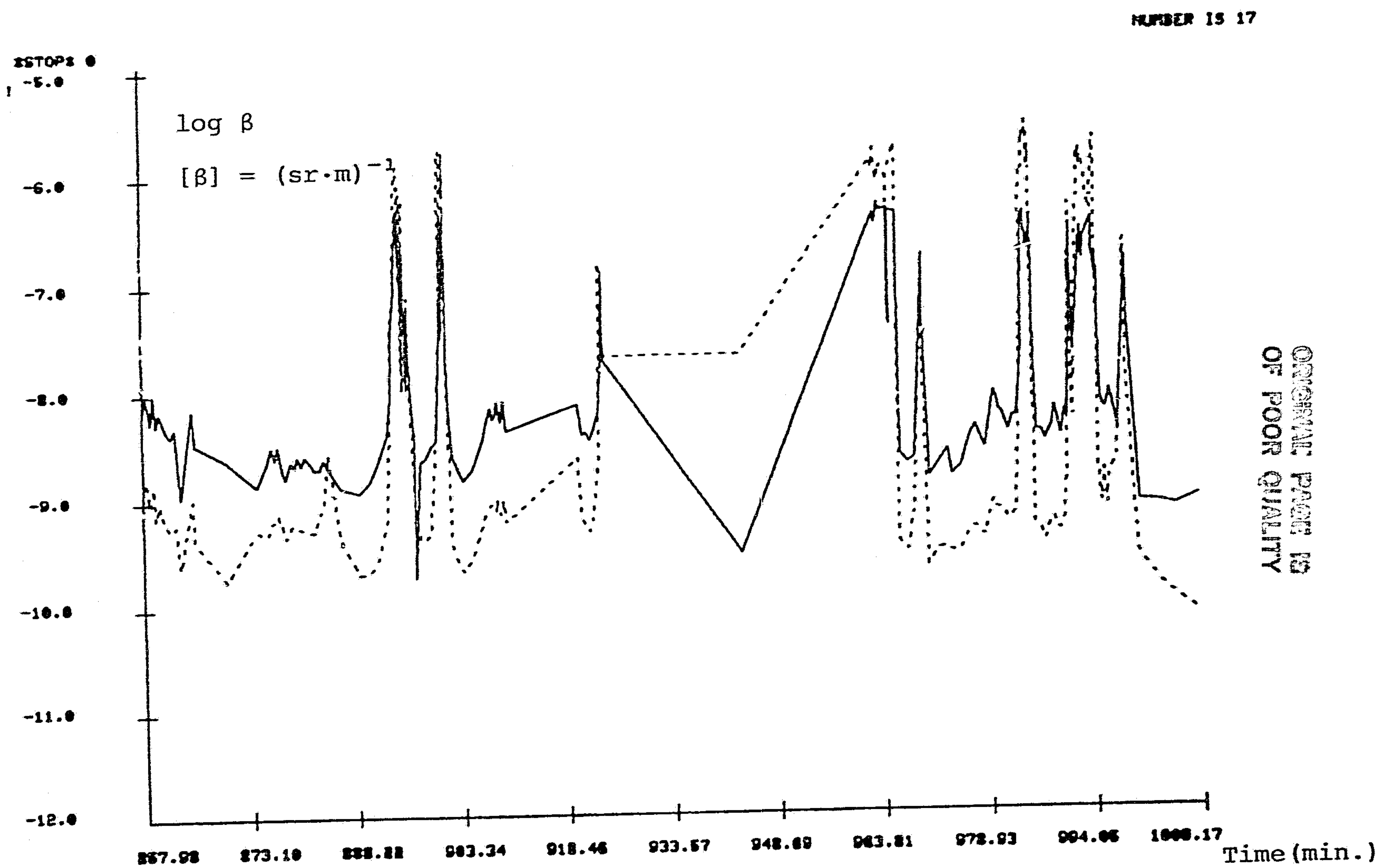


Fig. 21 Backscatter Coefficient β

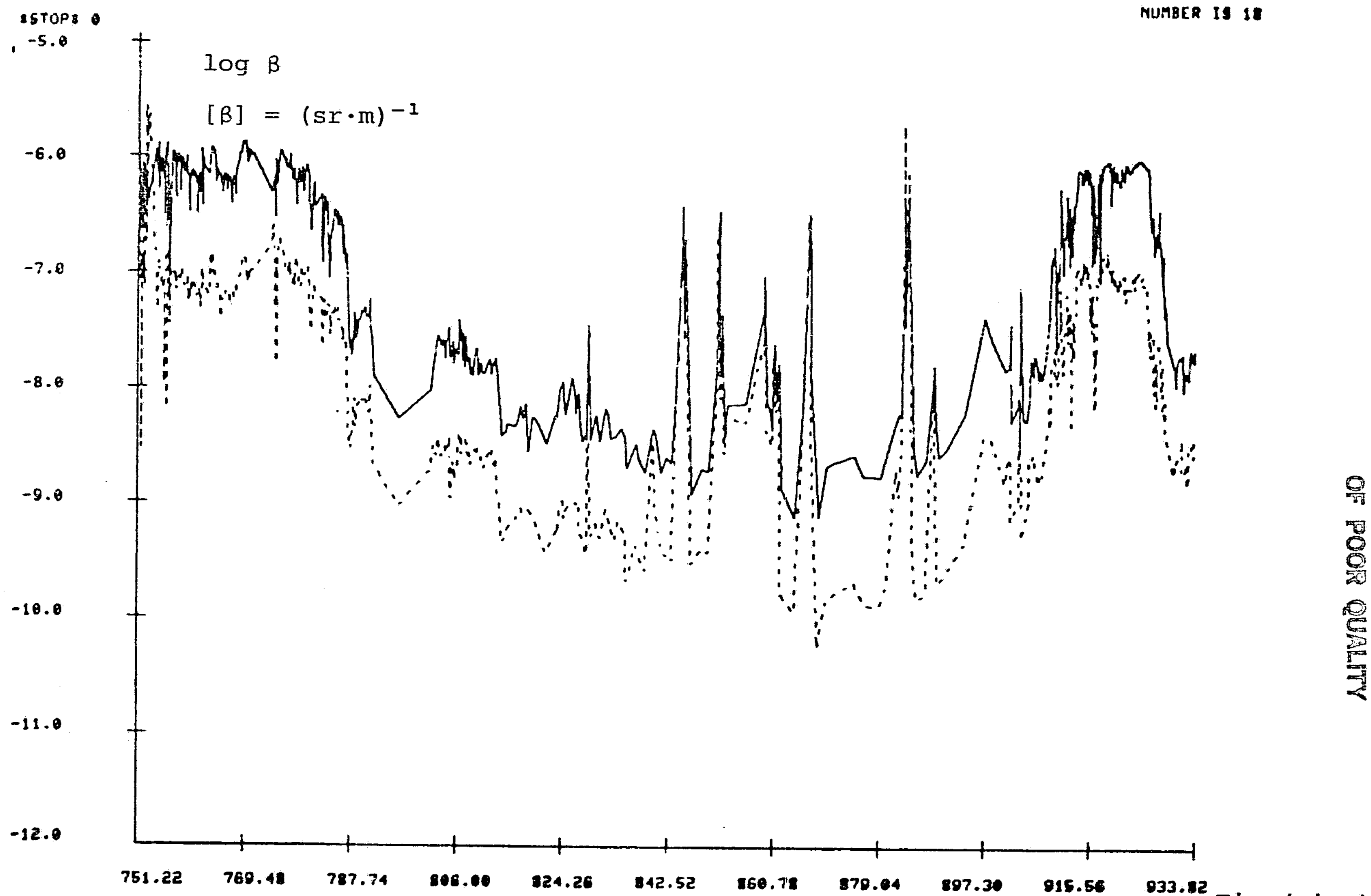


Fig. 22 Backscatter Coefficient β

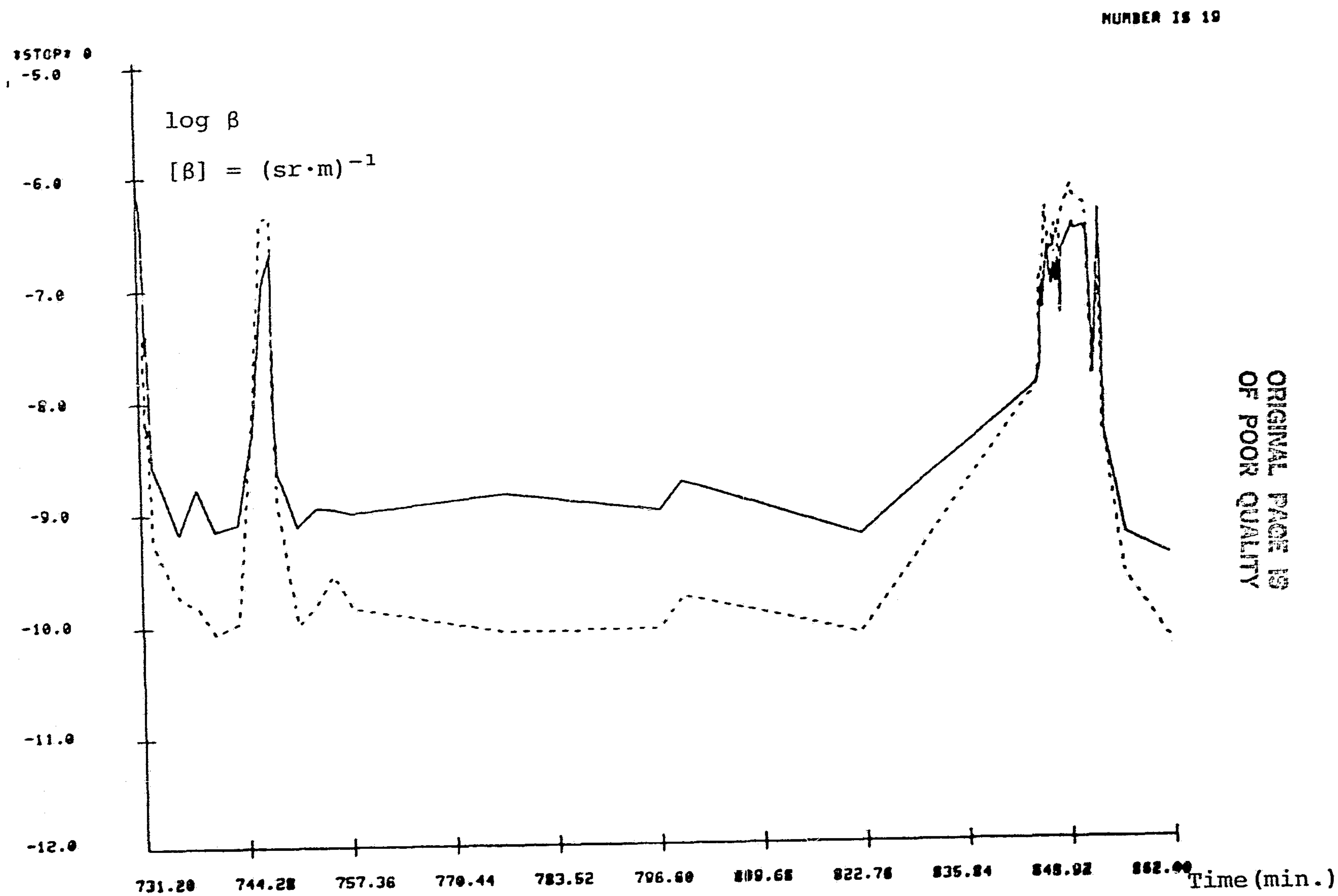


Fig. 23 Backscatter Coefficient β

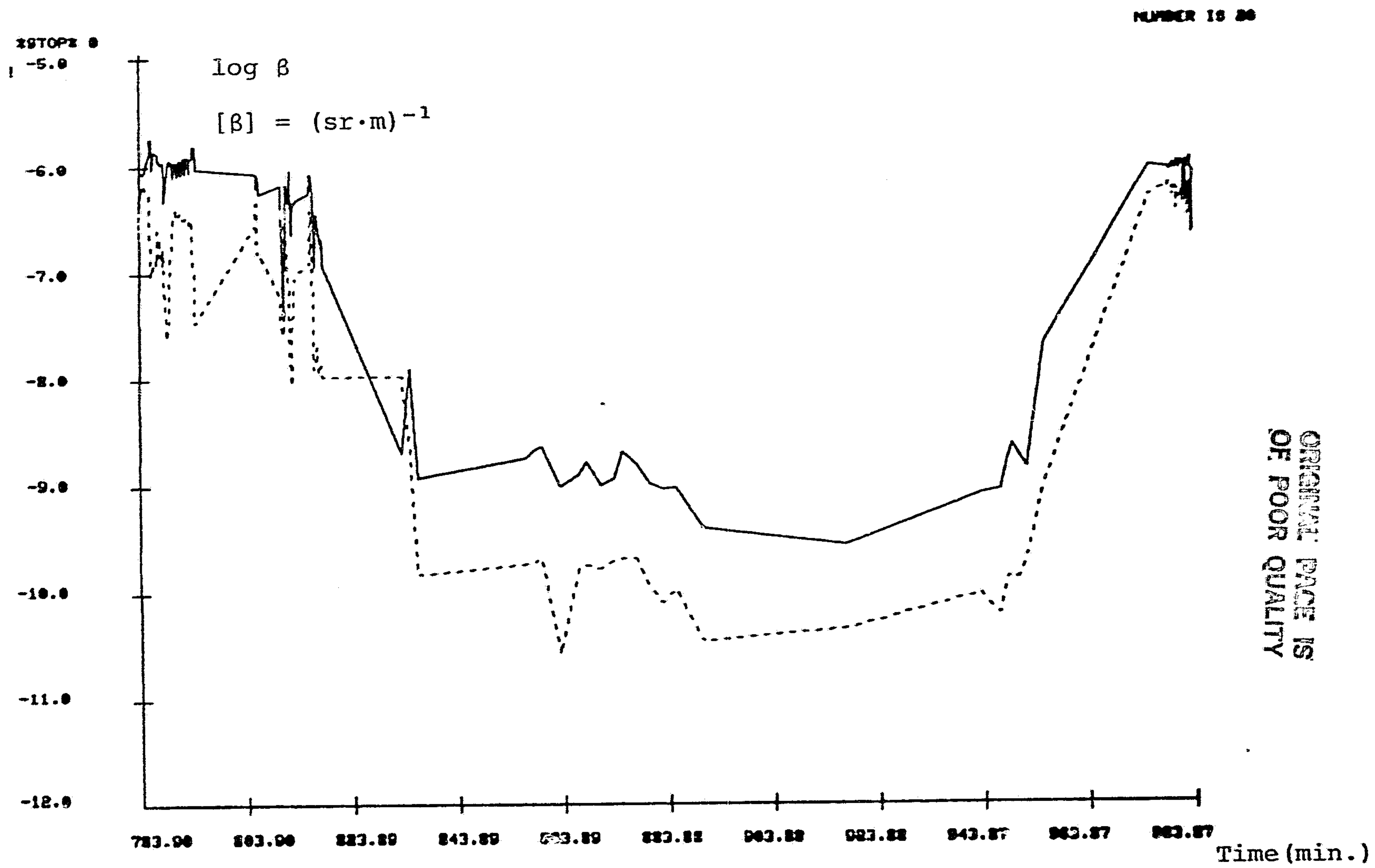


Fig. 24 Backscatter Coefficient β

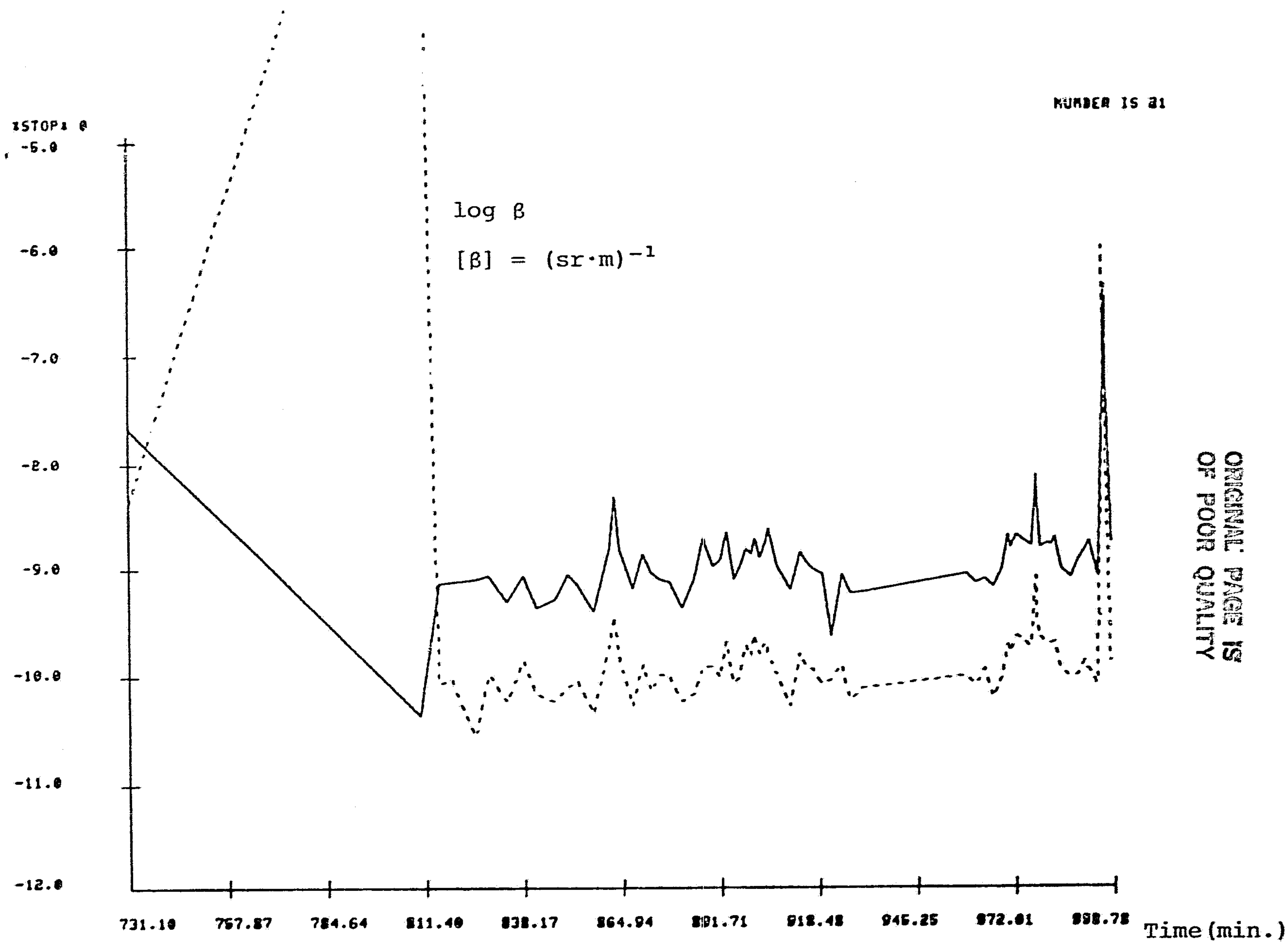


Fig. 25 Backscatter Coefficient β

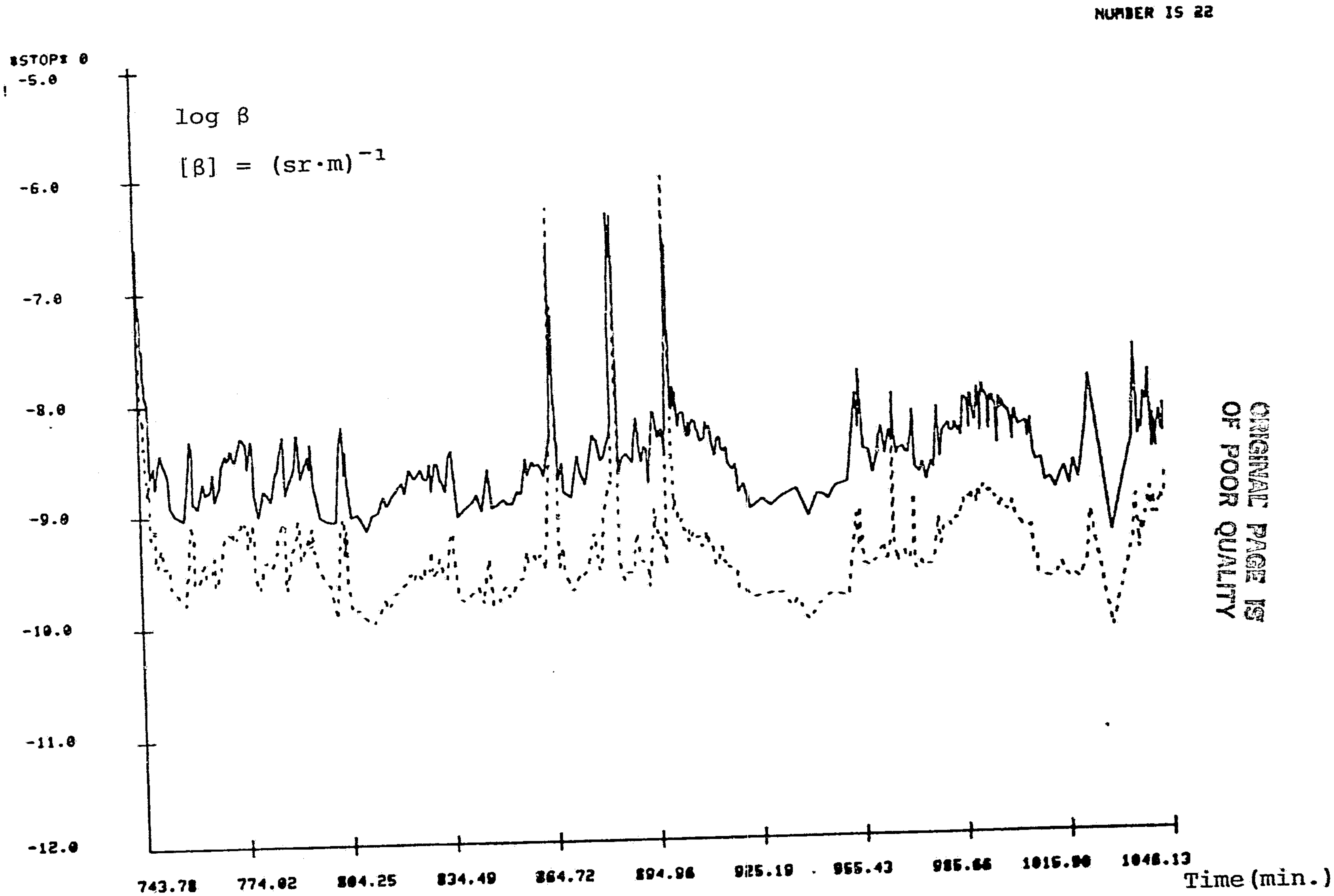


Fig. 26 Backscatter Coefficient β

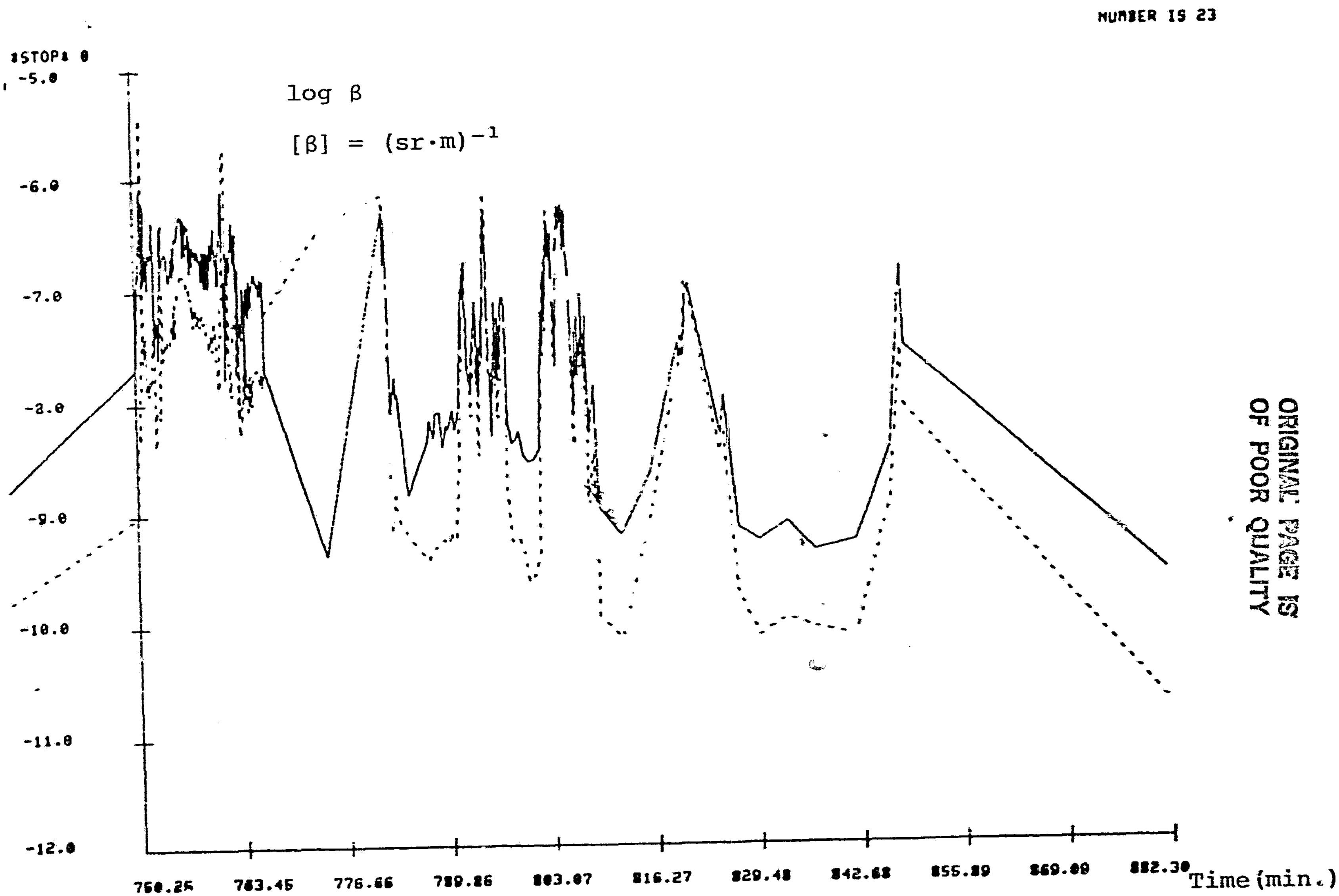


Fig. 27 Backscatter Coefficient β

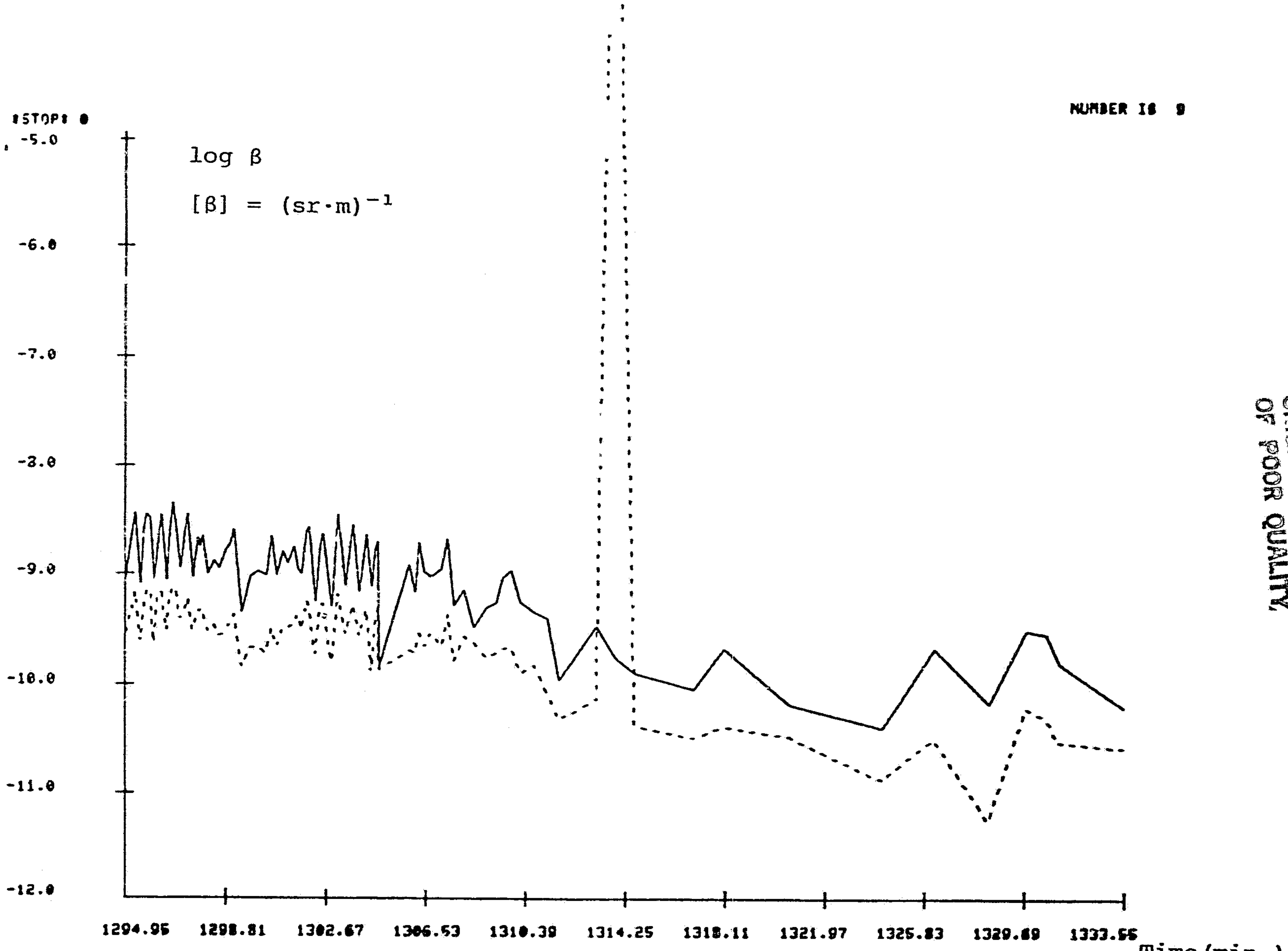


Fig. 27-A Backscatter Coefficient β

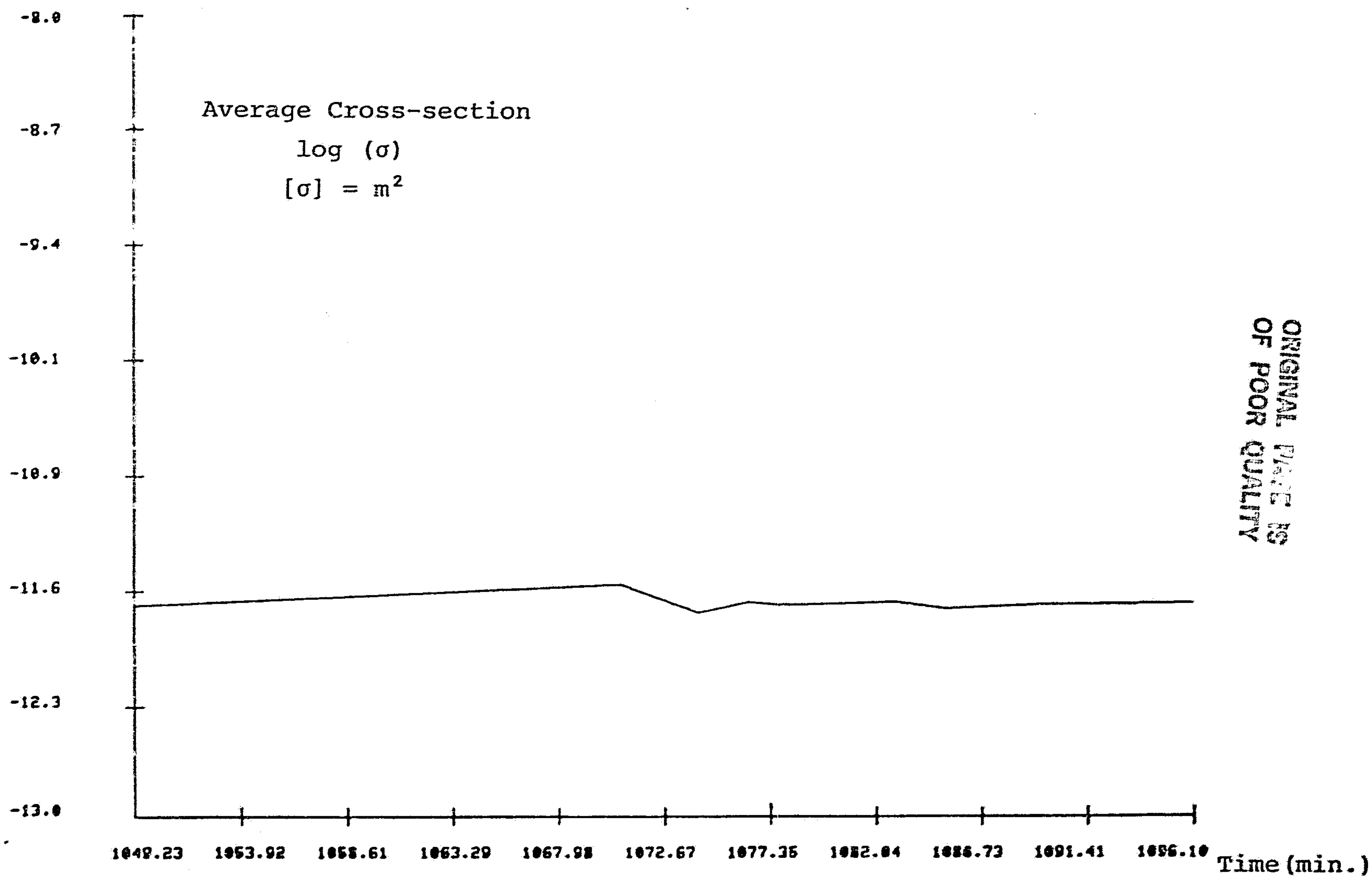


Fig. 28 Average Cross-section σ

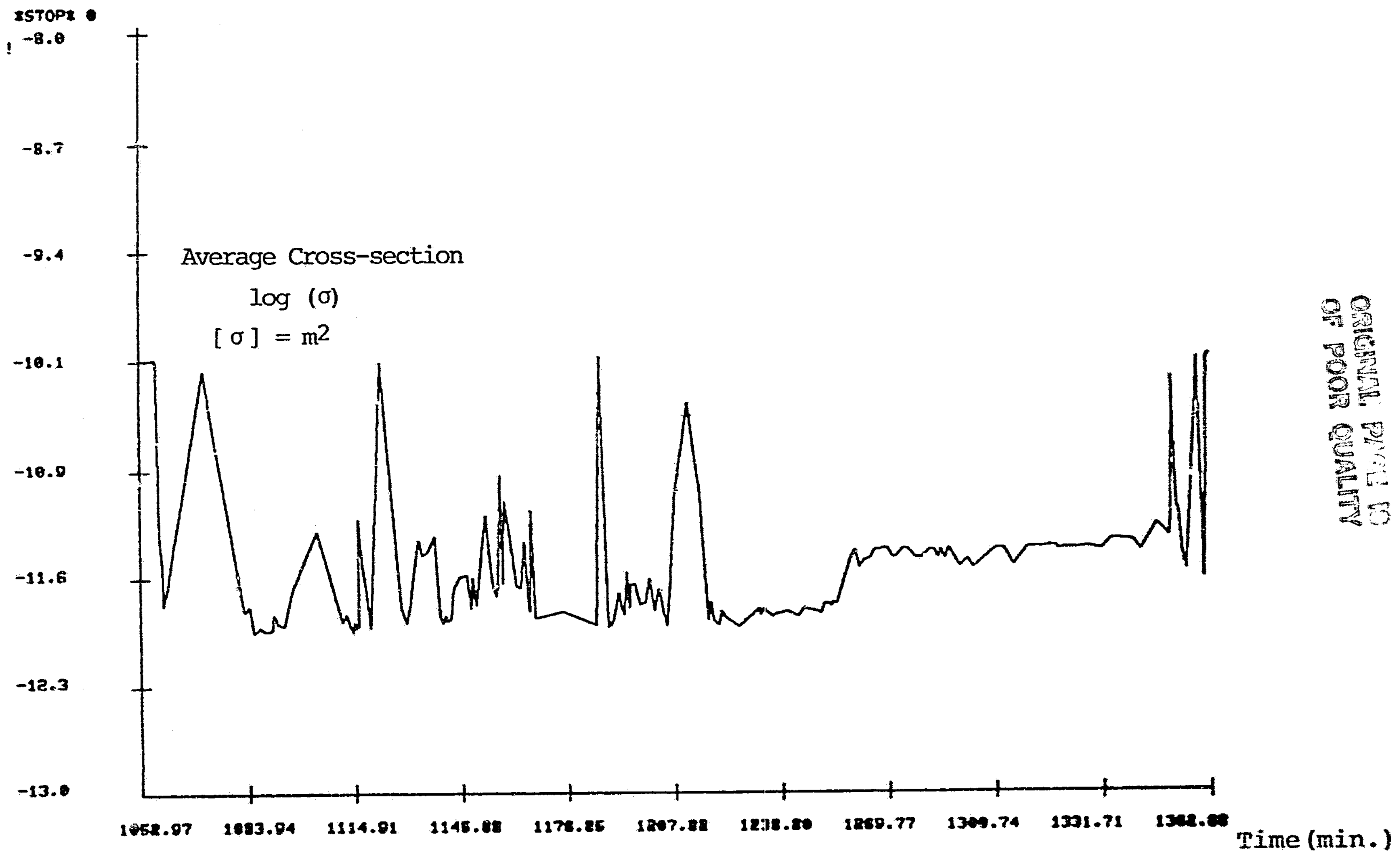


Fig.29 Average Cross-section σ

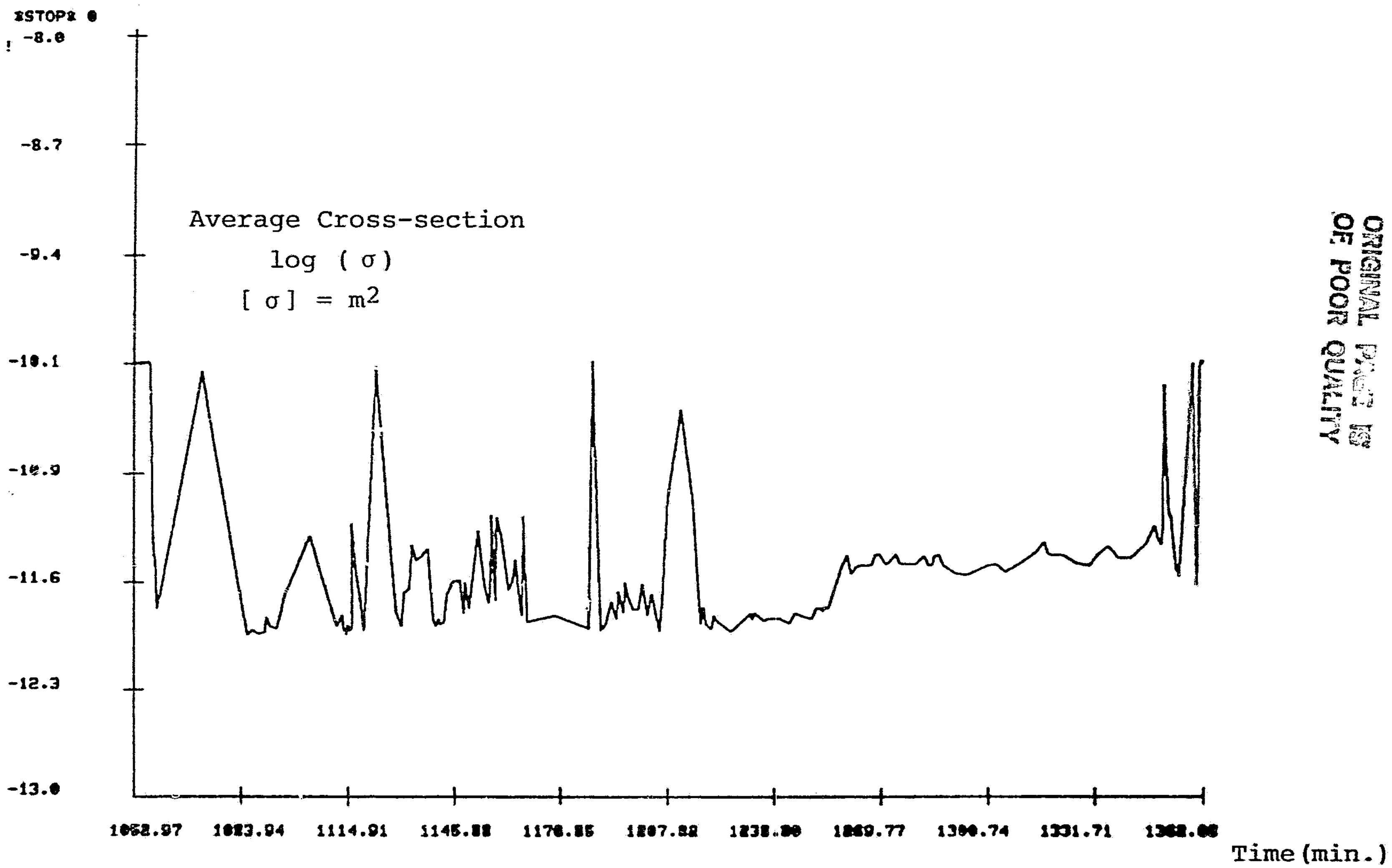


Fig.30 Average Cross Section σ

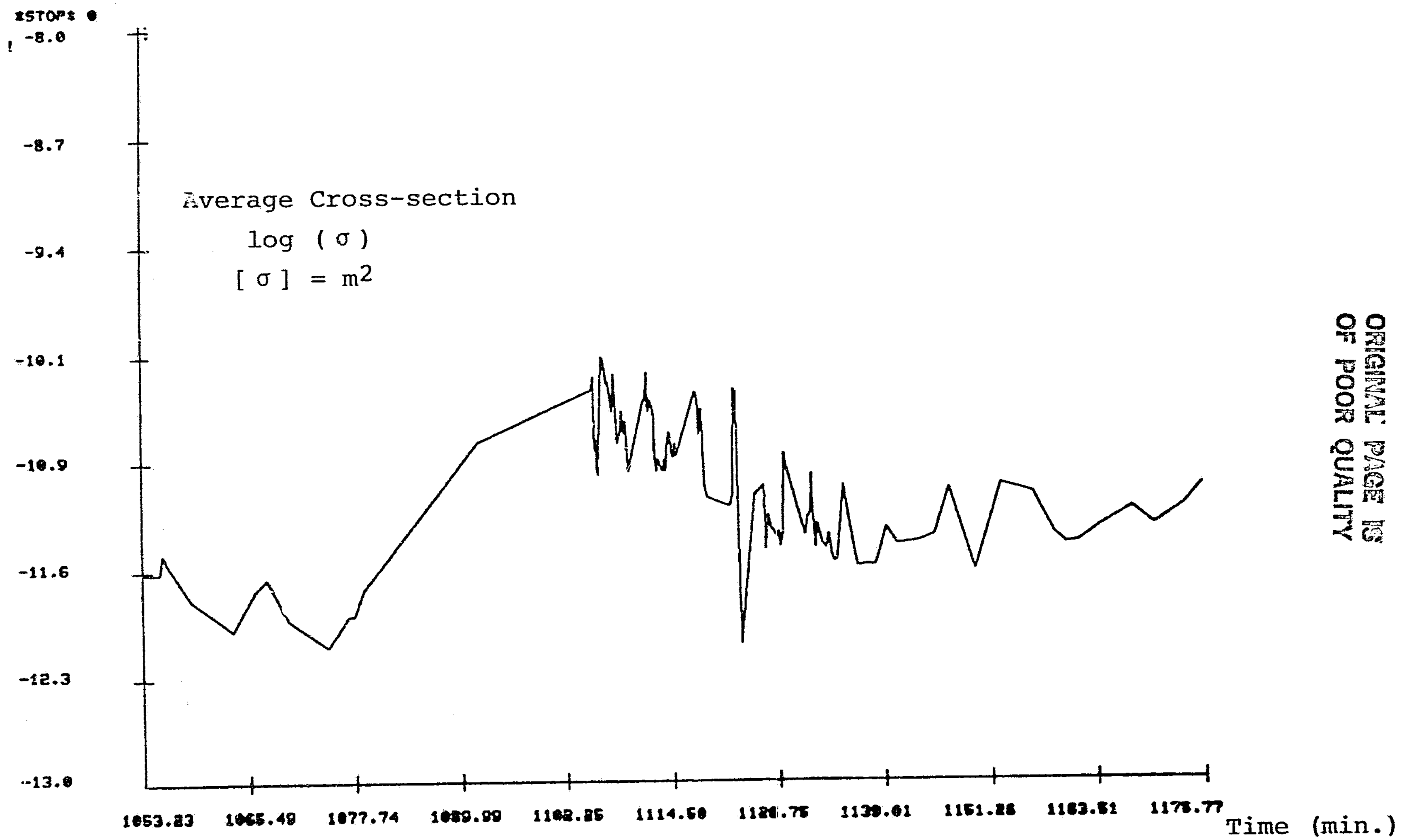


Fig.31 Average Cross-section σ

ORIGINAL PAGE IS
OF POOR QUALITY

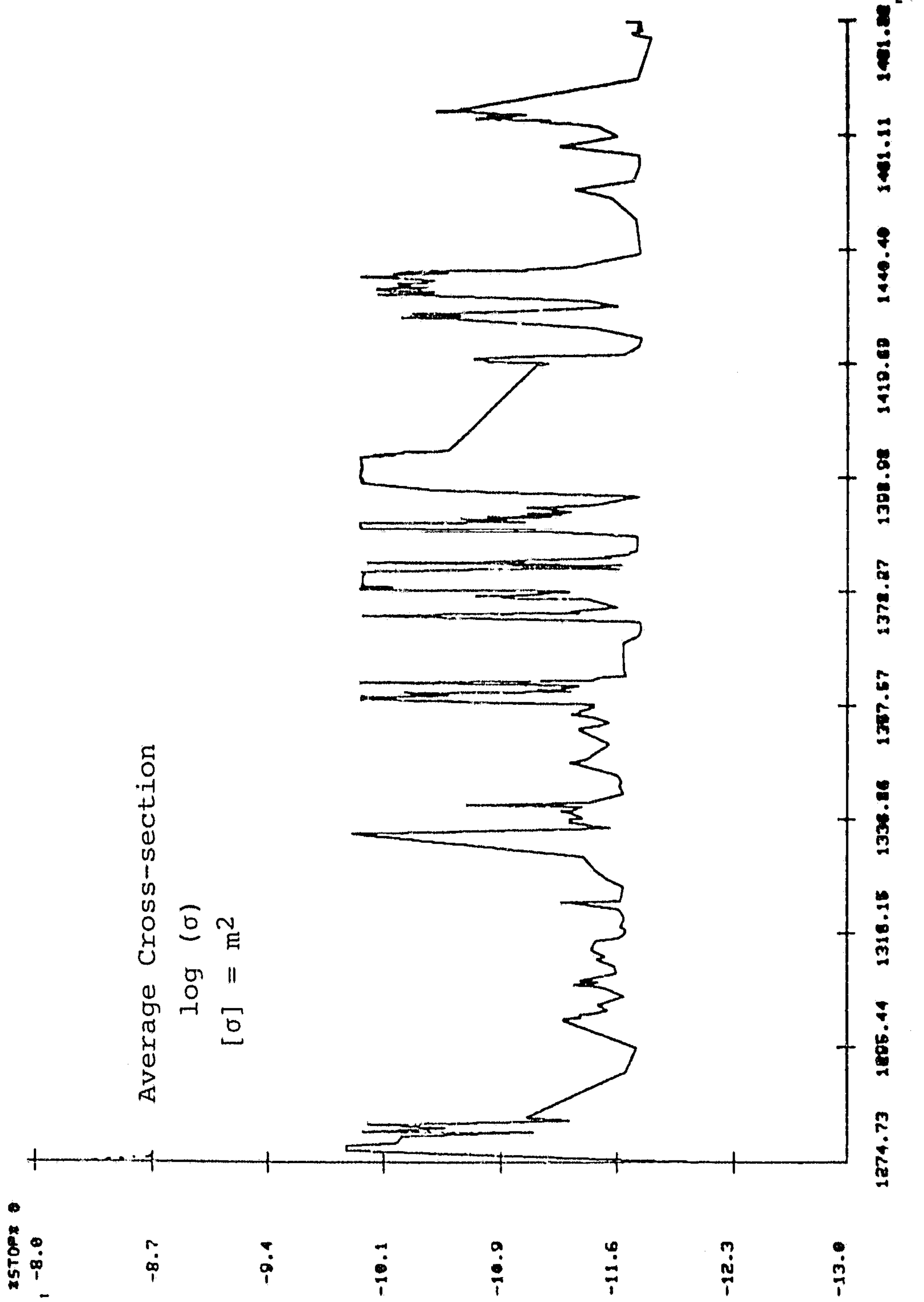


Fig. 32 Average Cross-section σ

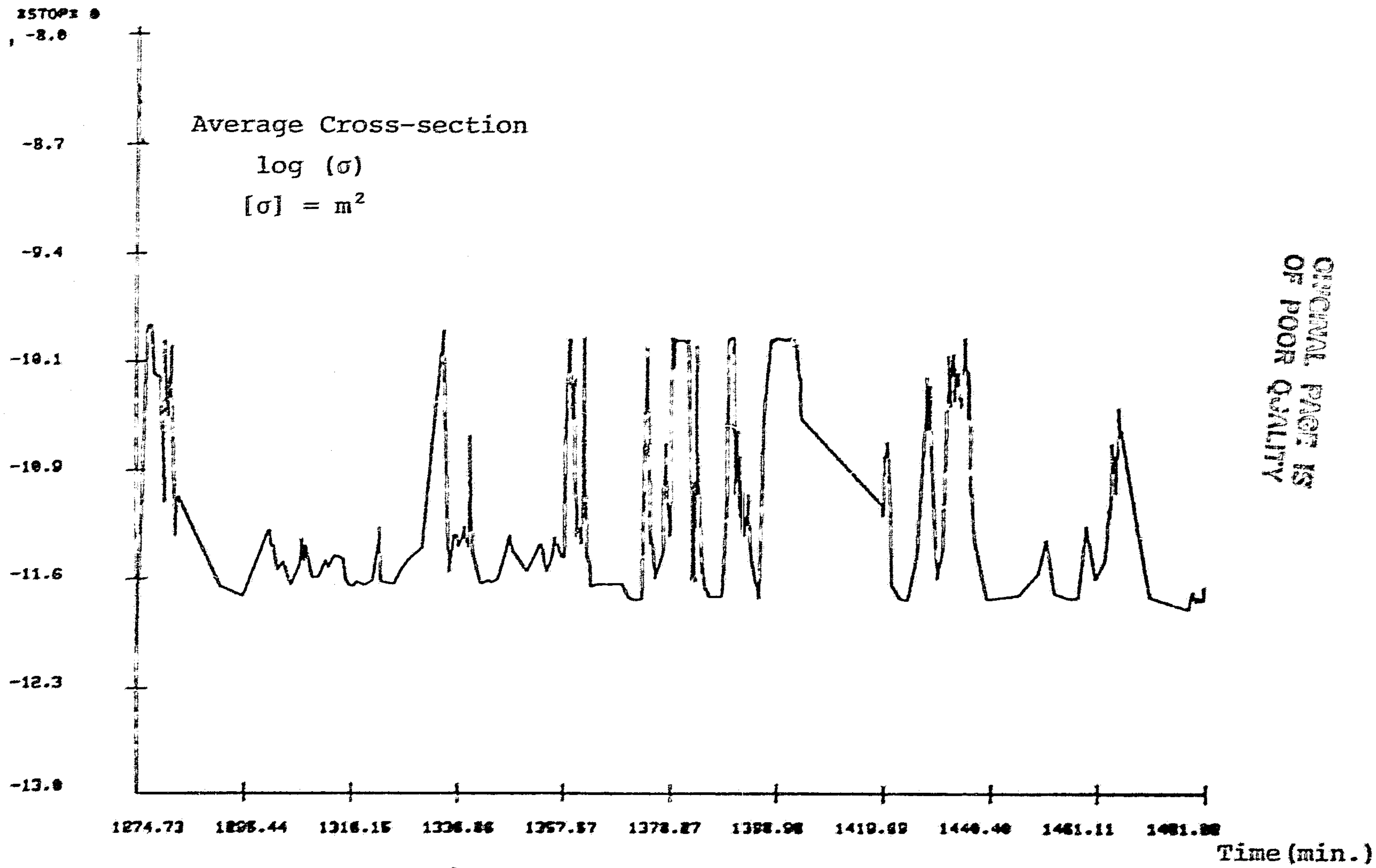


Fig. 33 Average Cross-section σ

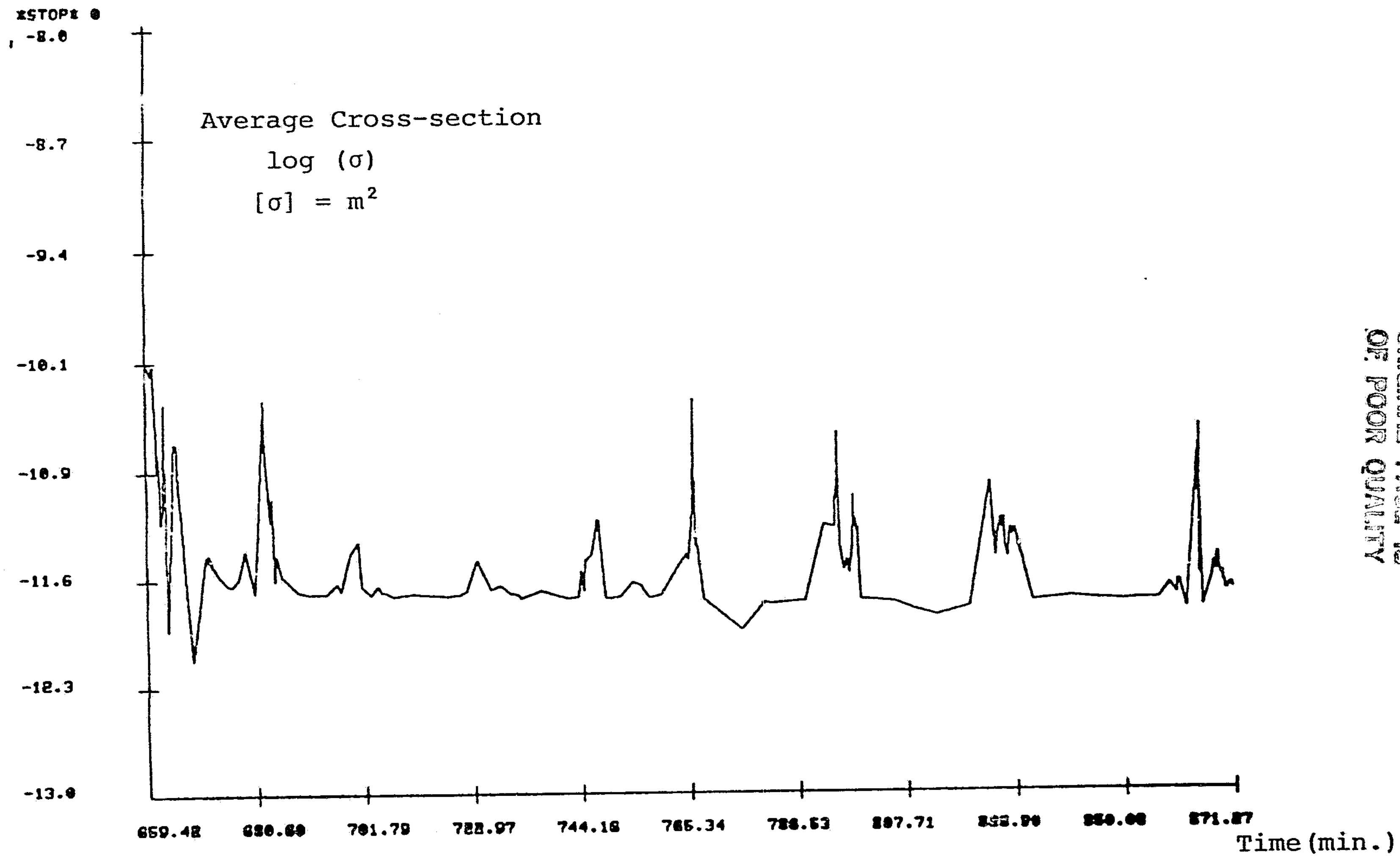


Fig. 34 Average Cross-section σ

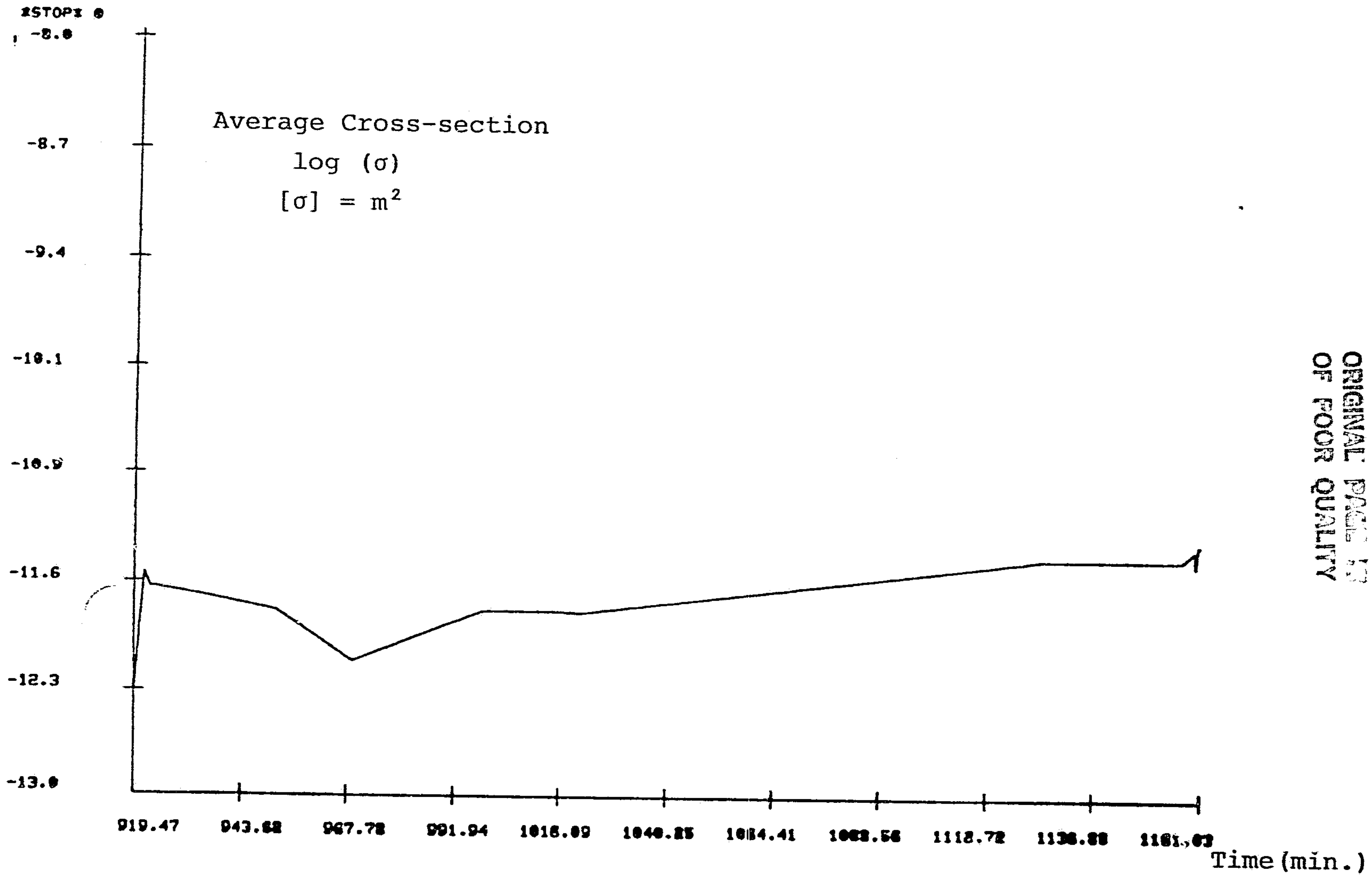


Fig. 35 Average Cross-section σ

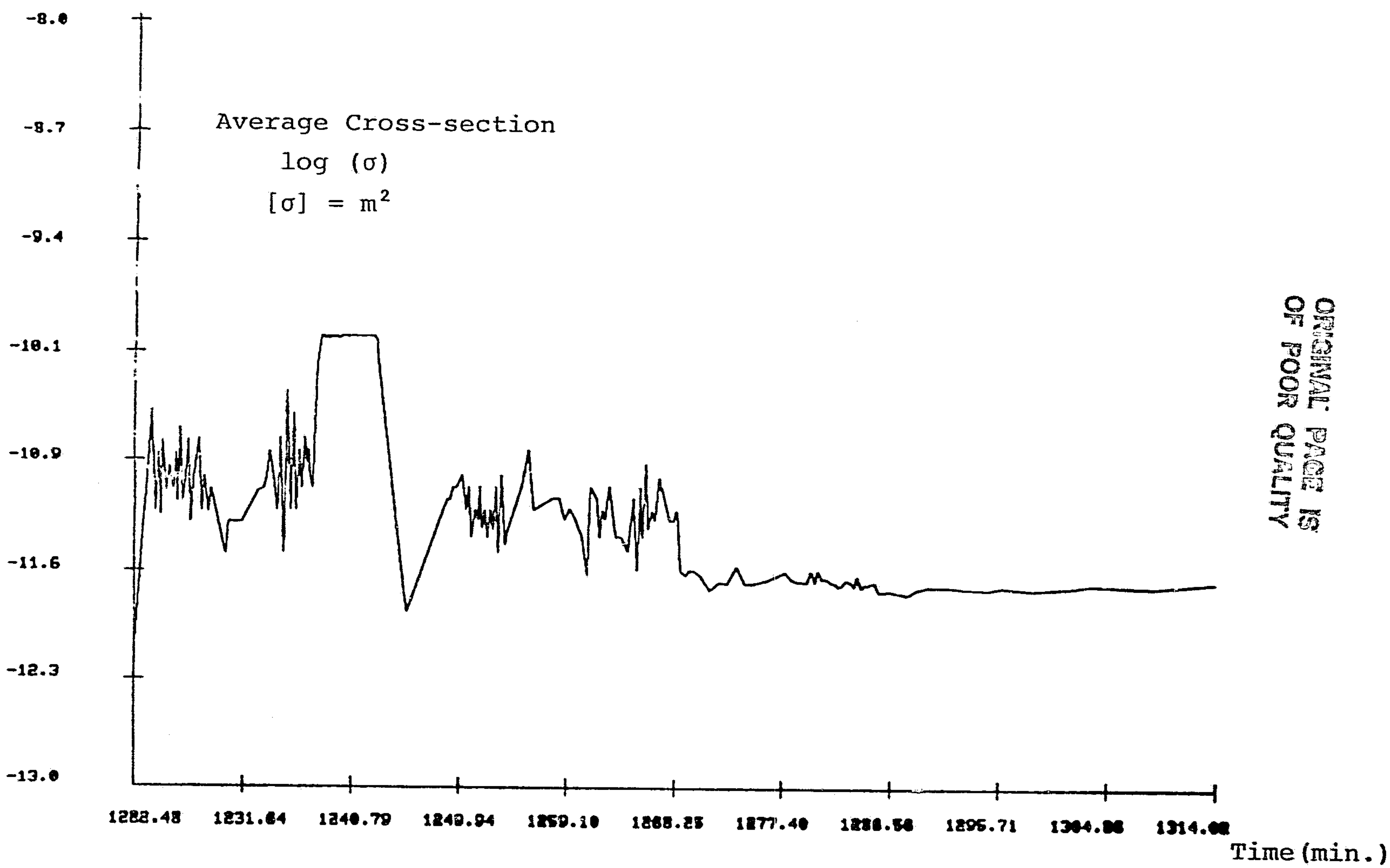


Fig. 36 Average Cross-section σ

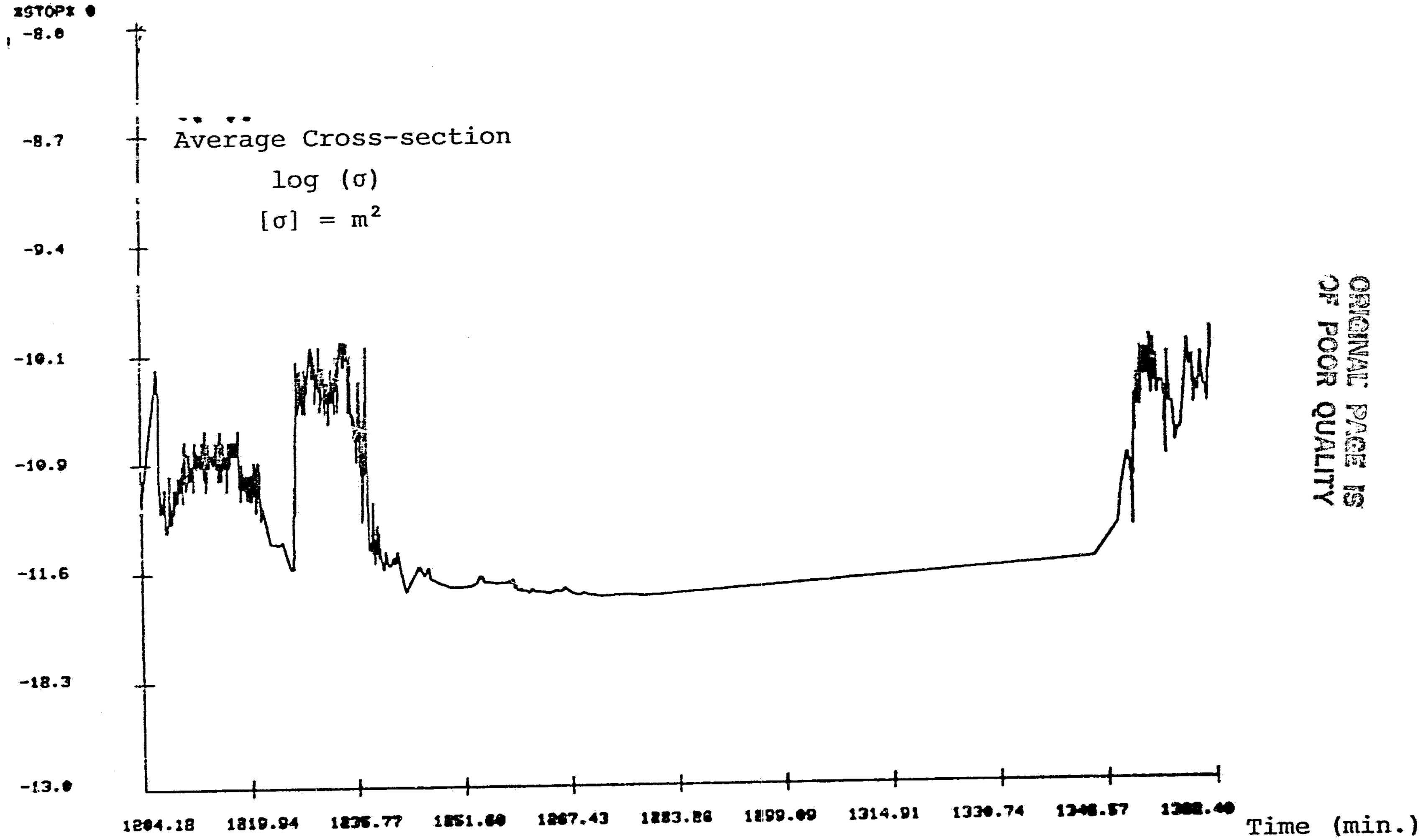


Fig. 37 Average Cross-section σ

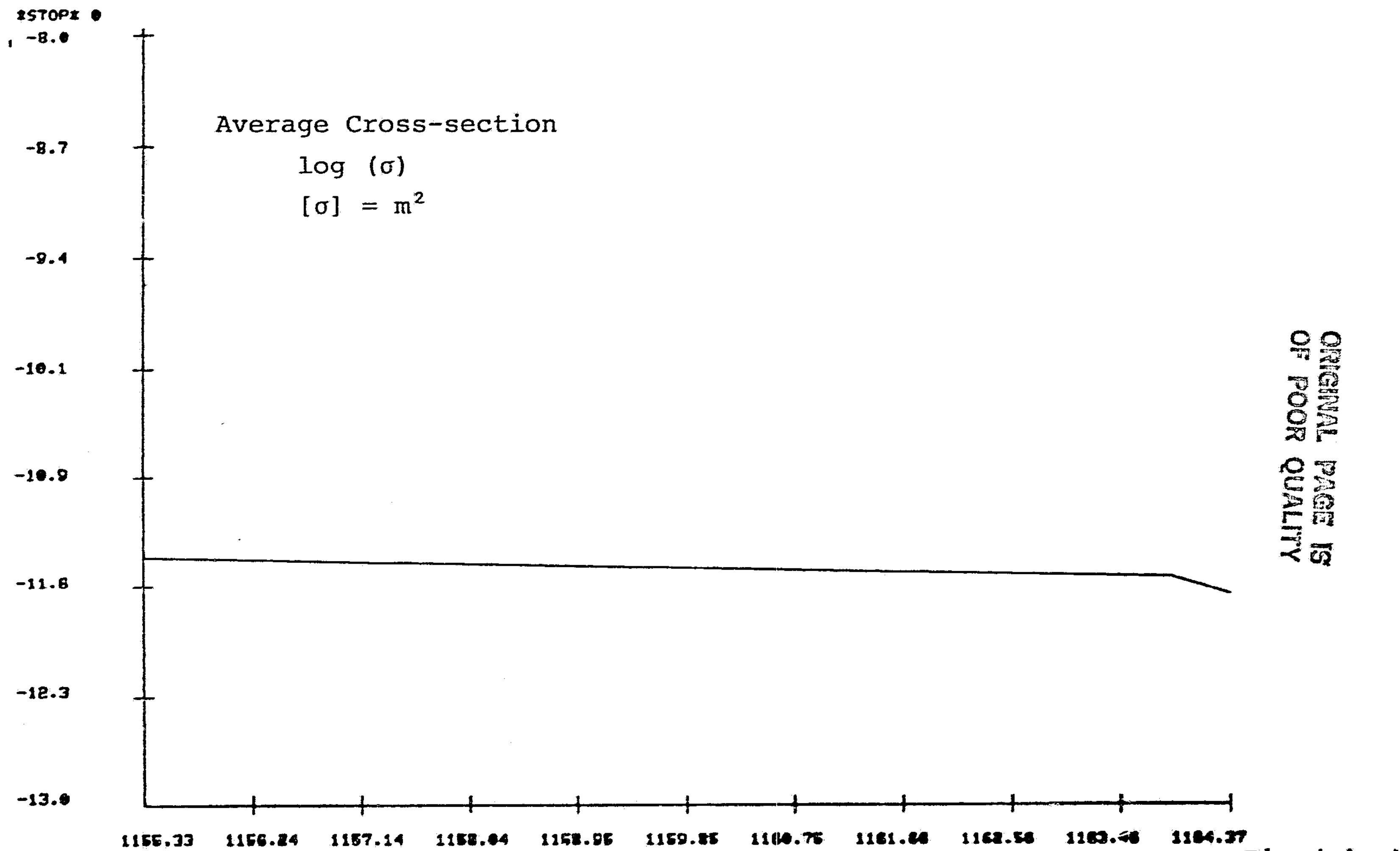


Fig. 37 Average Cross-section σ

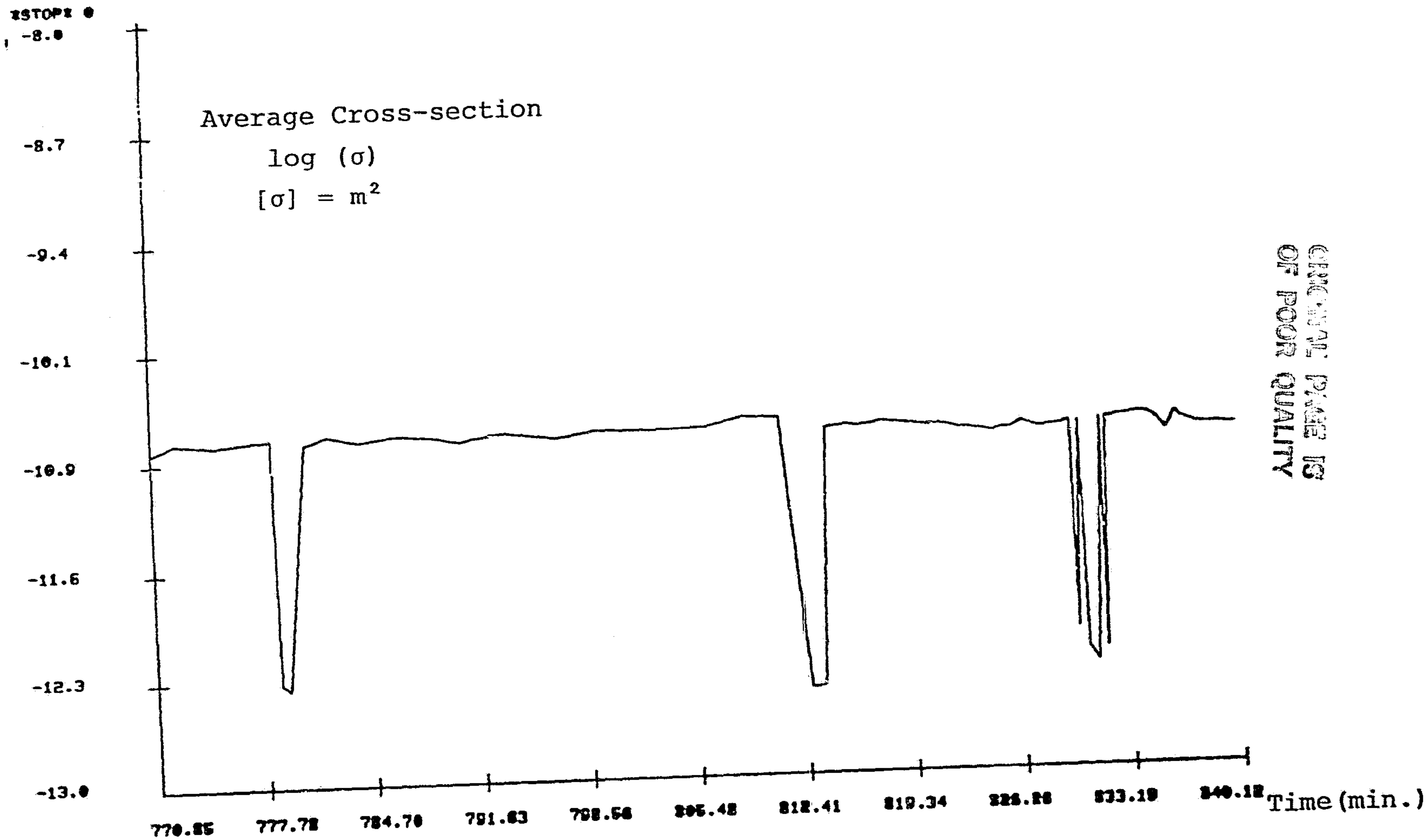


Fig. 39 Average Cross-section σ

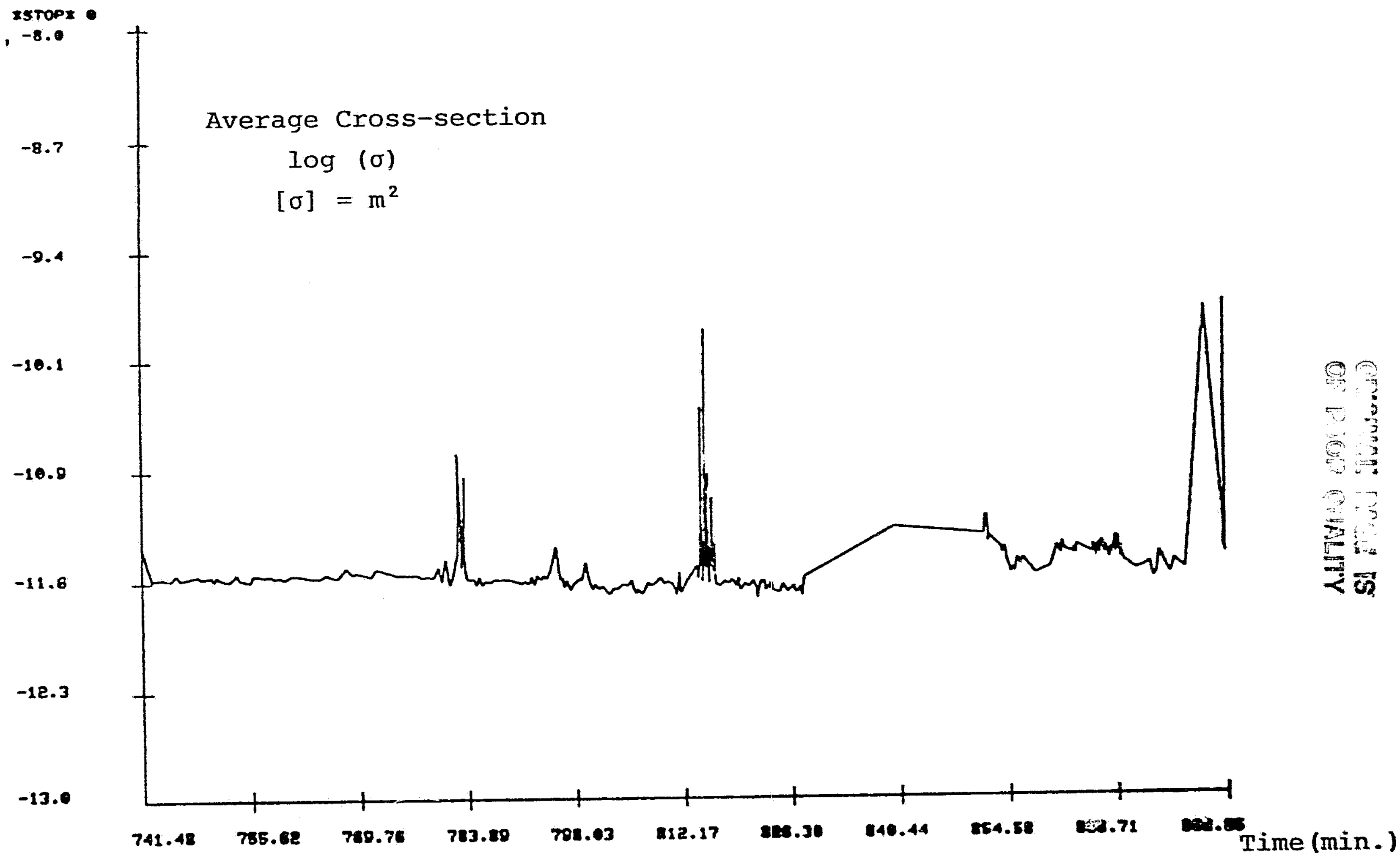
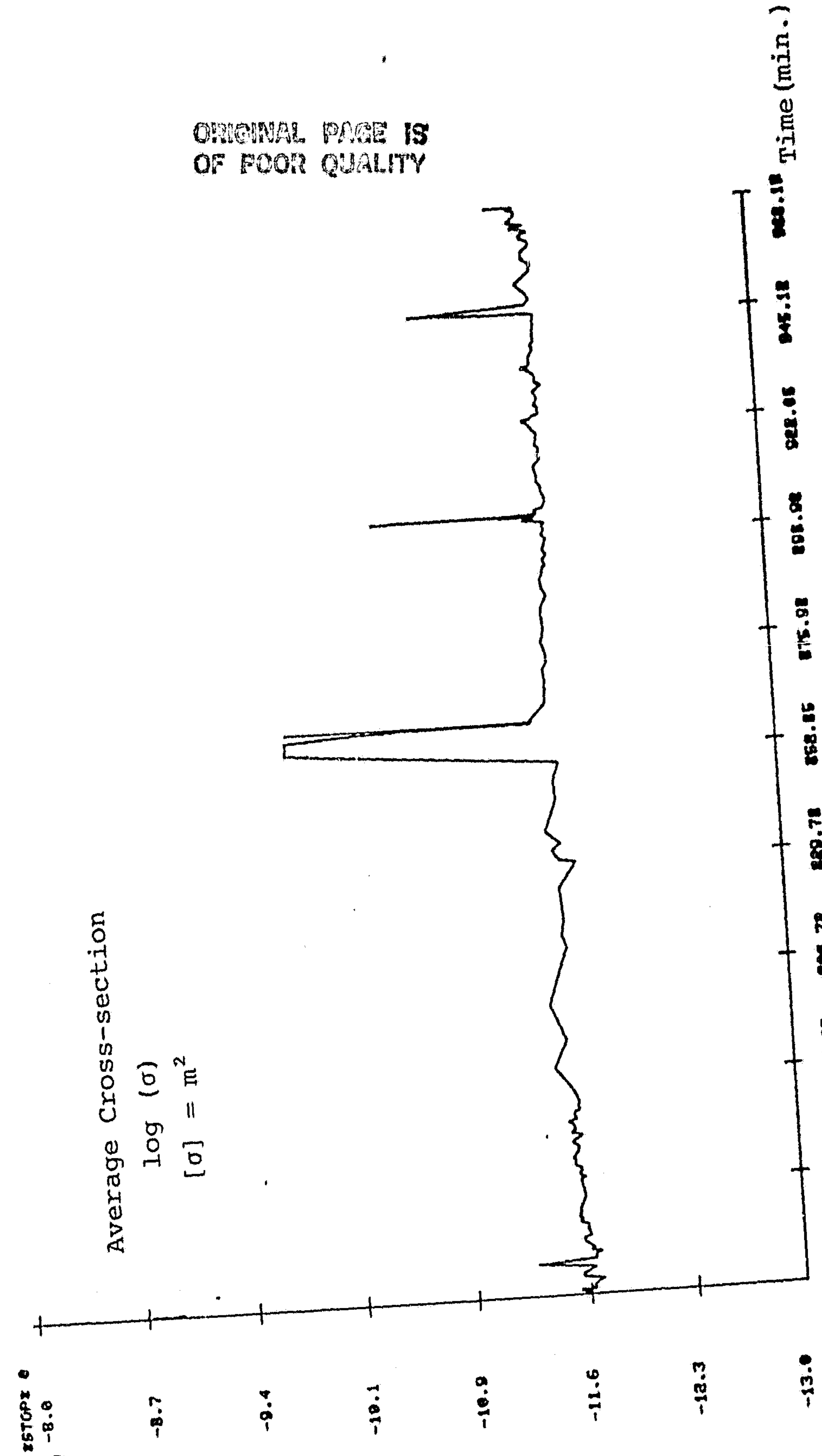


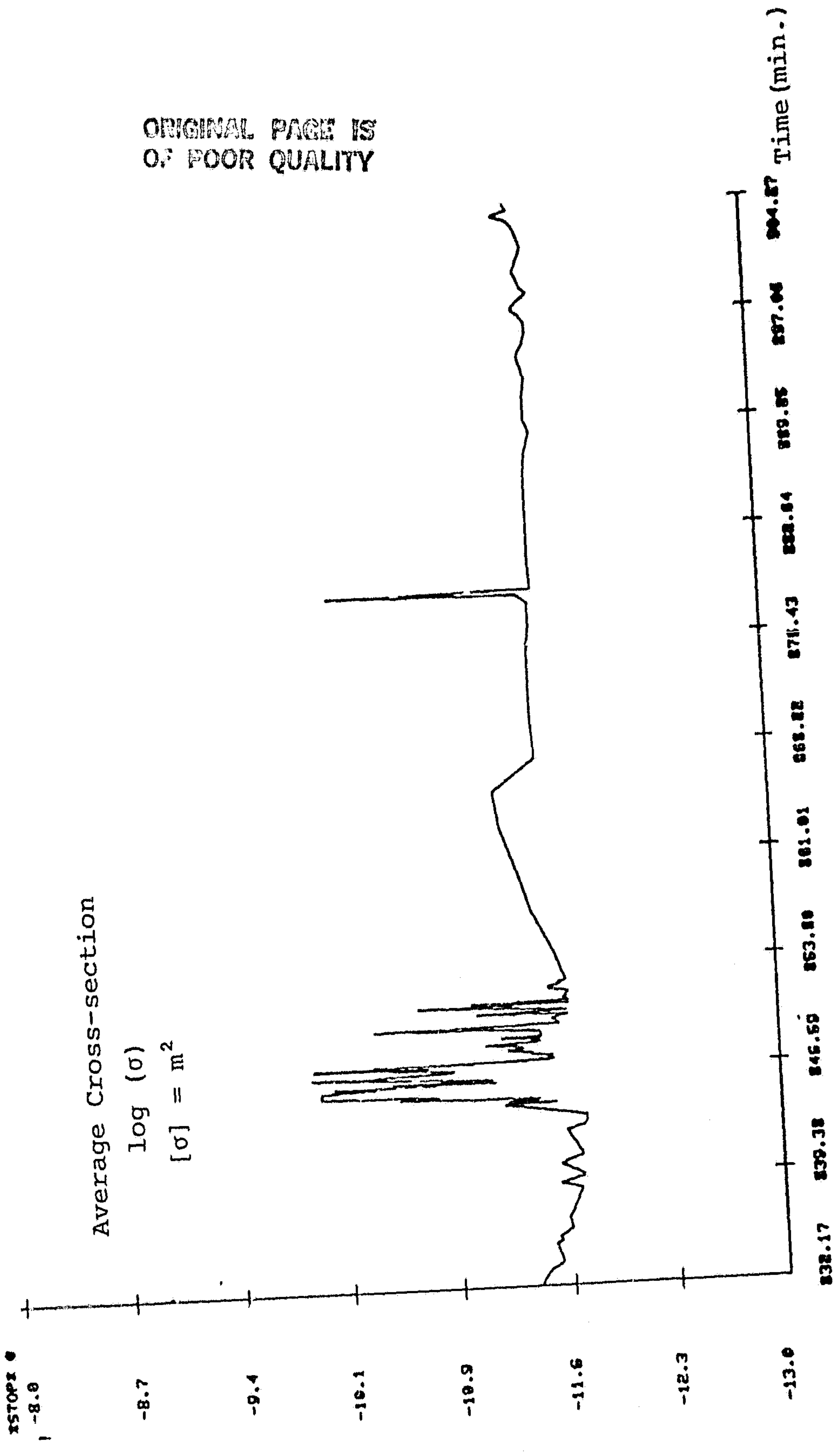
Fig. 40 Average Cross-section σ

ORIGINAL PAGE IS
OF POOR QUALITY



Average Cross-section σ

Fig. 41



Average Cross-section σ

Fig. 42

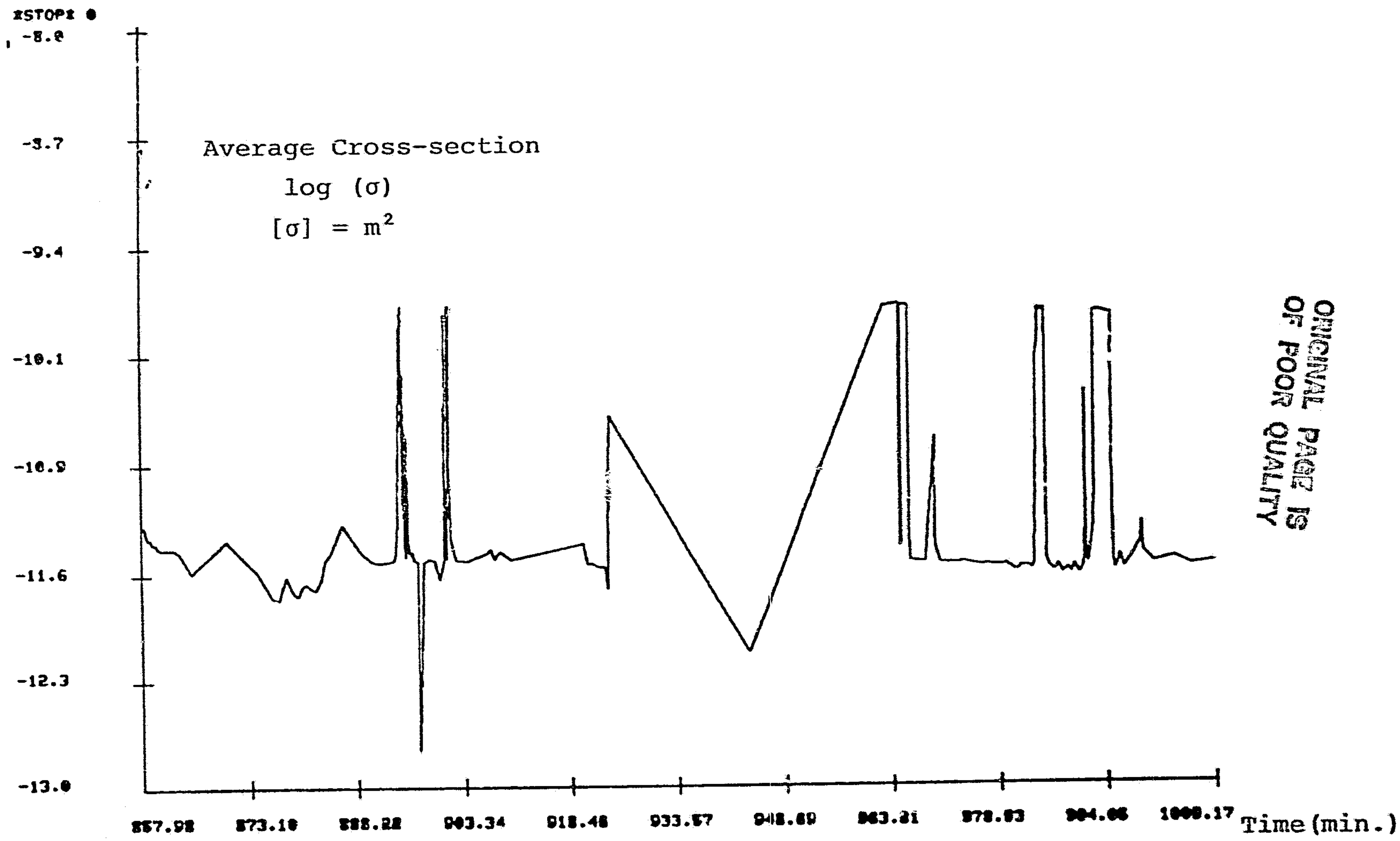


Fig. 43 Average Cross-section σ

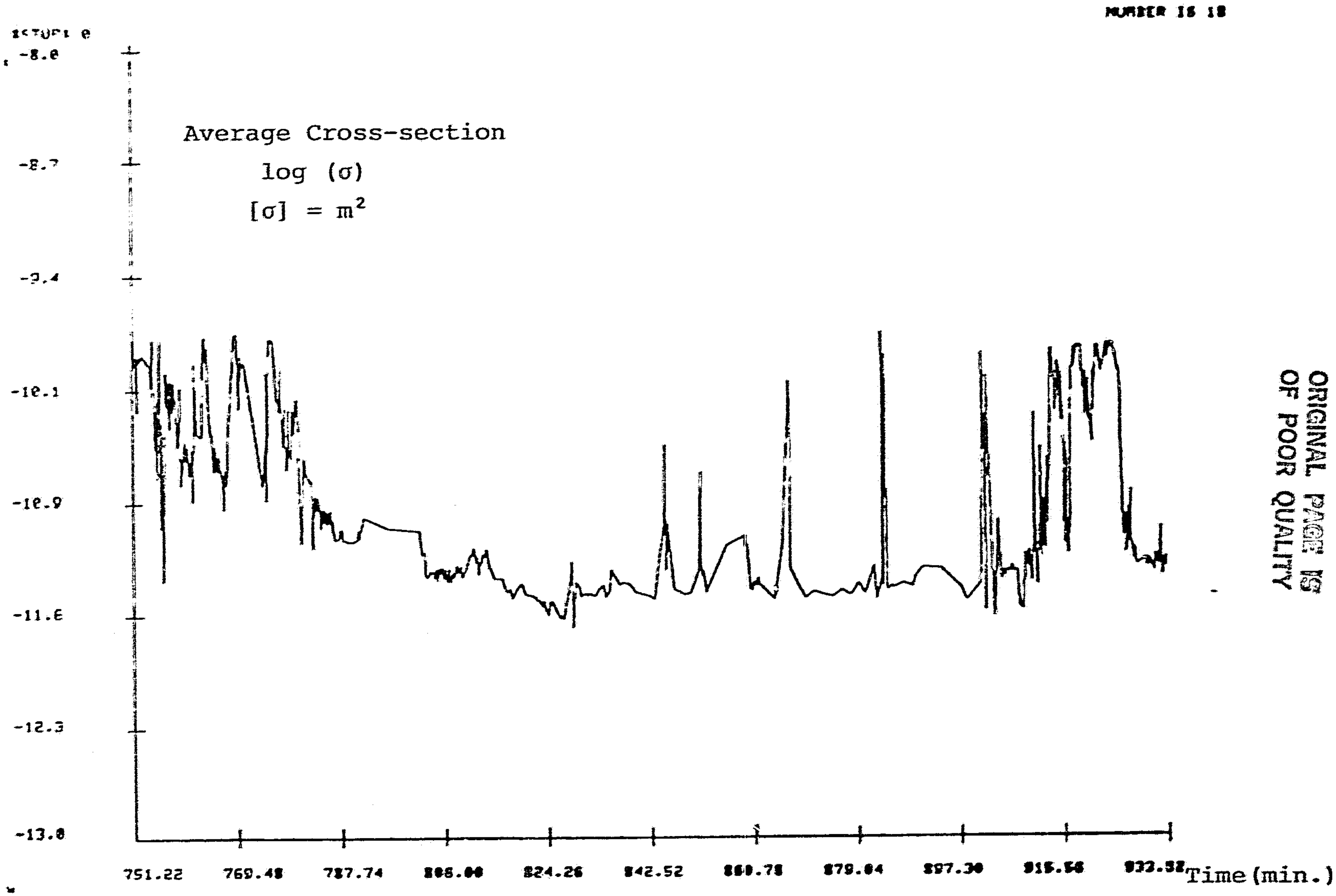


Fig. 44 Average Cross-section σ

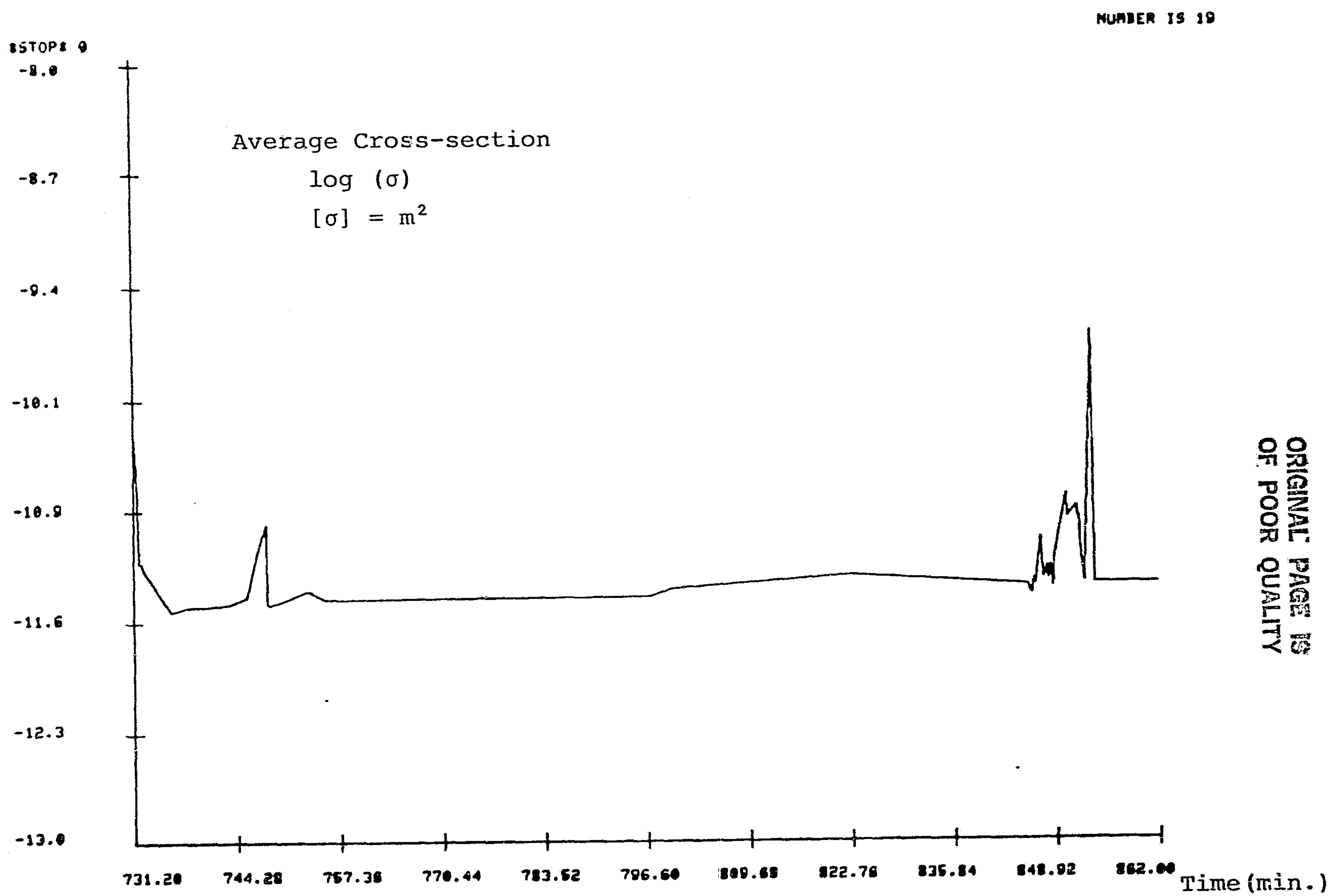


Fig. 45 Average Cross-section σ

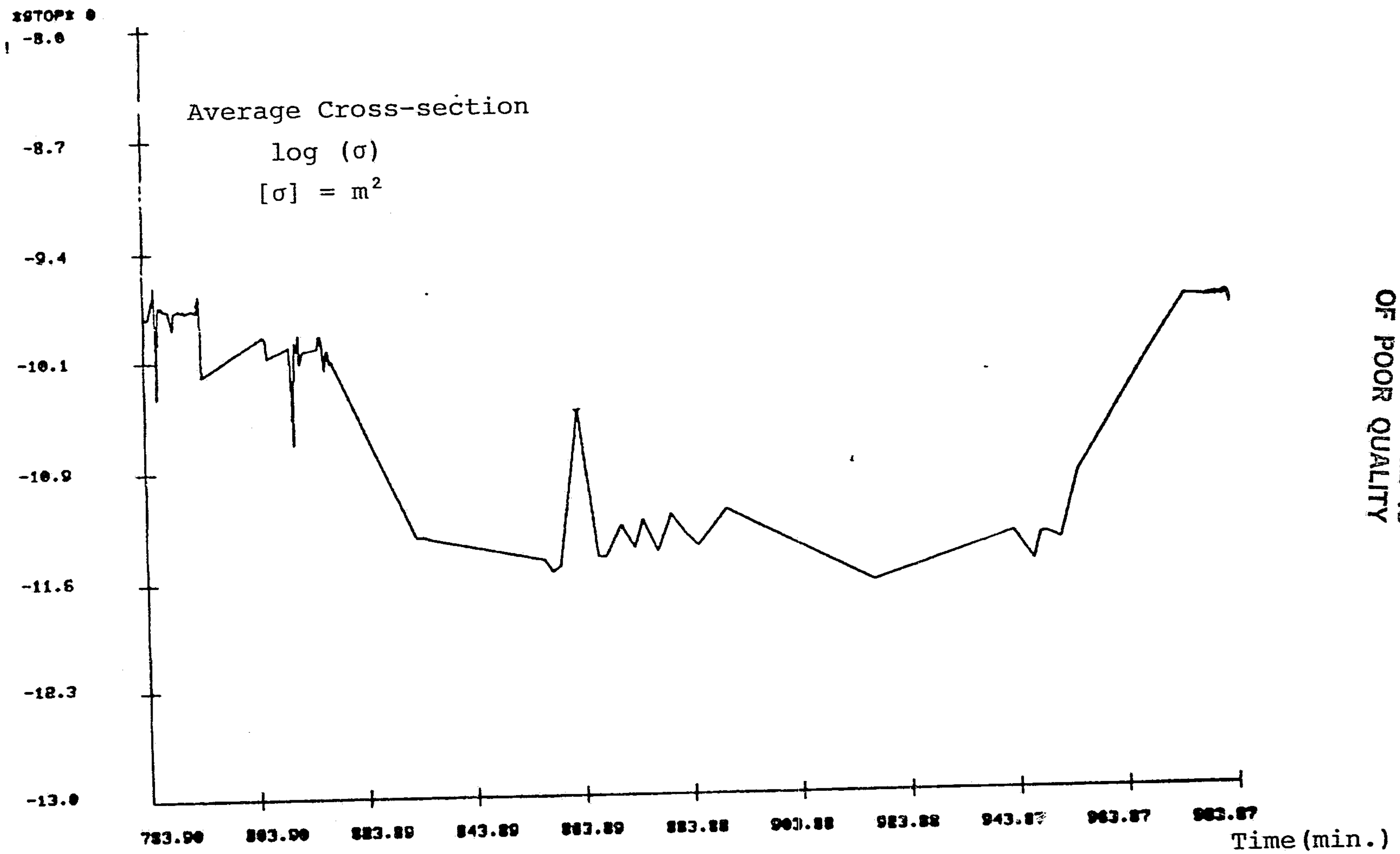


Fig. 46 Average Cross-section σ

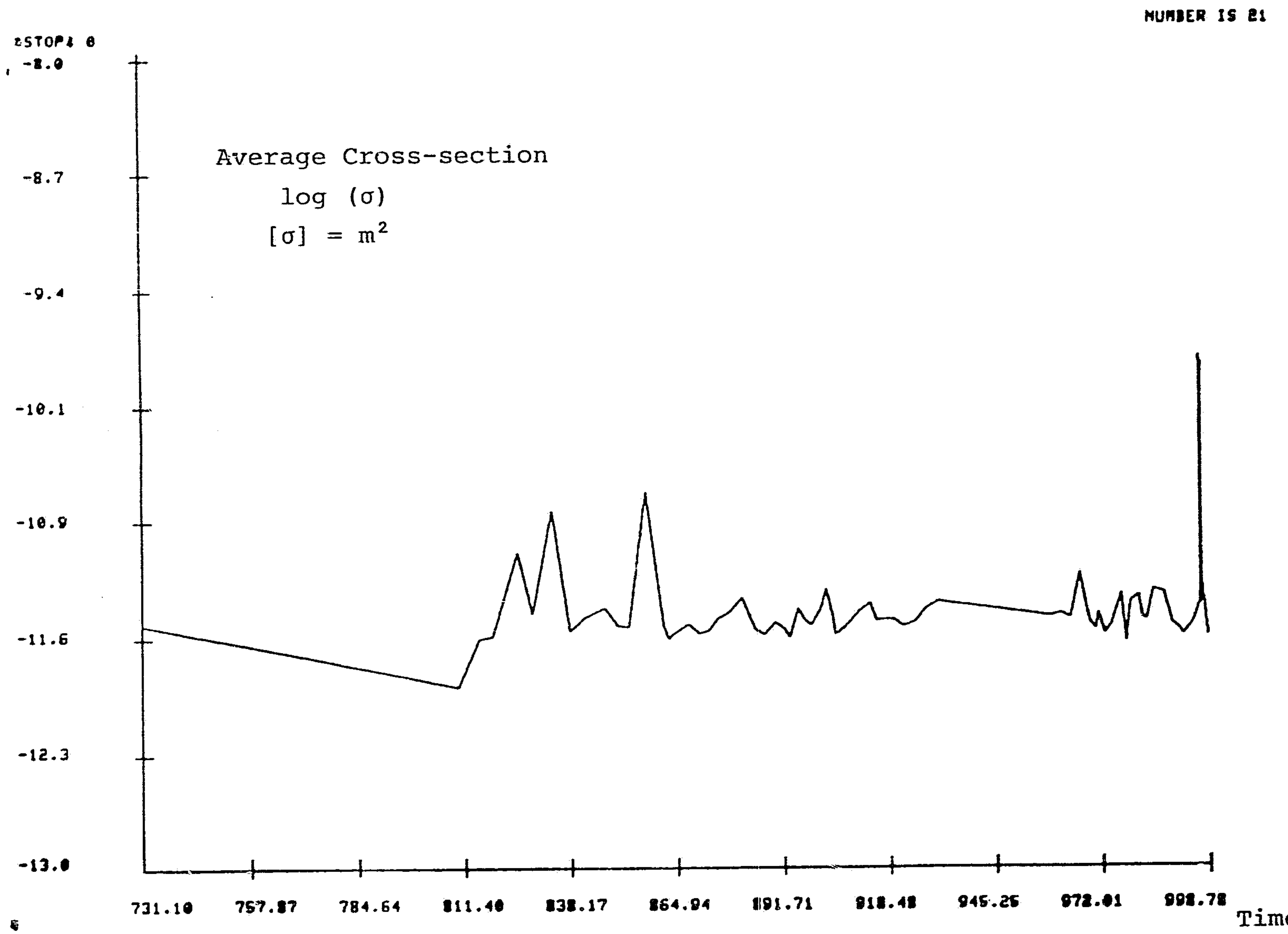


Fig. 47 Average Cross-section σ

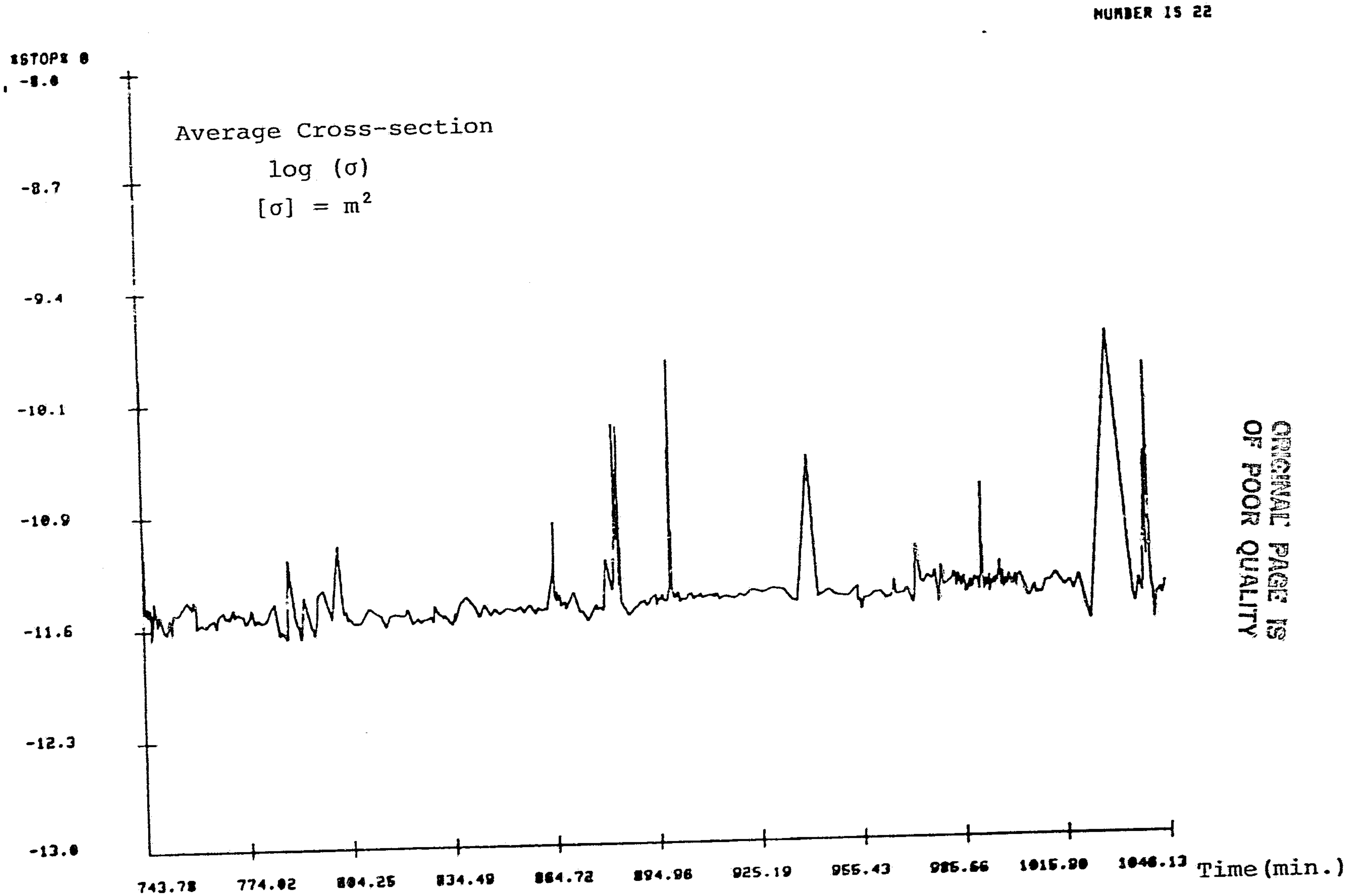


Fig. 48 Average Cross-section σ

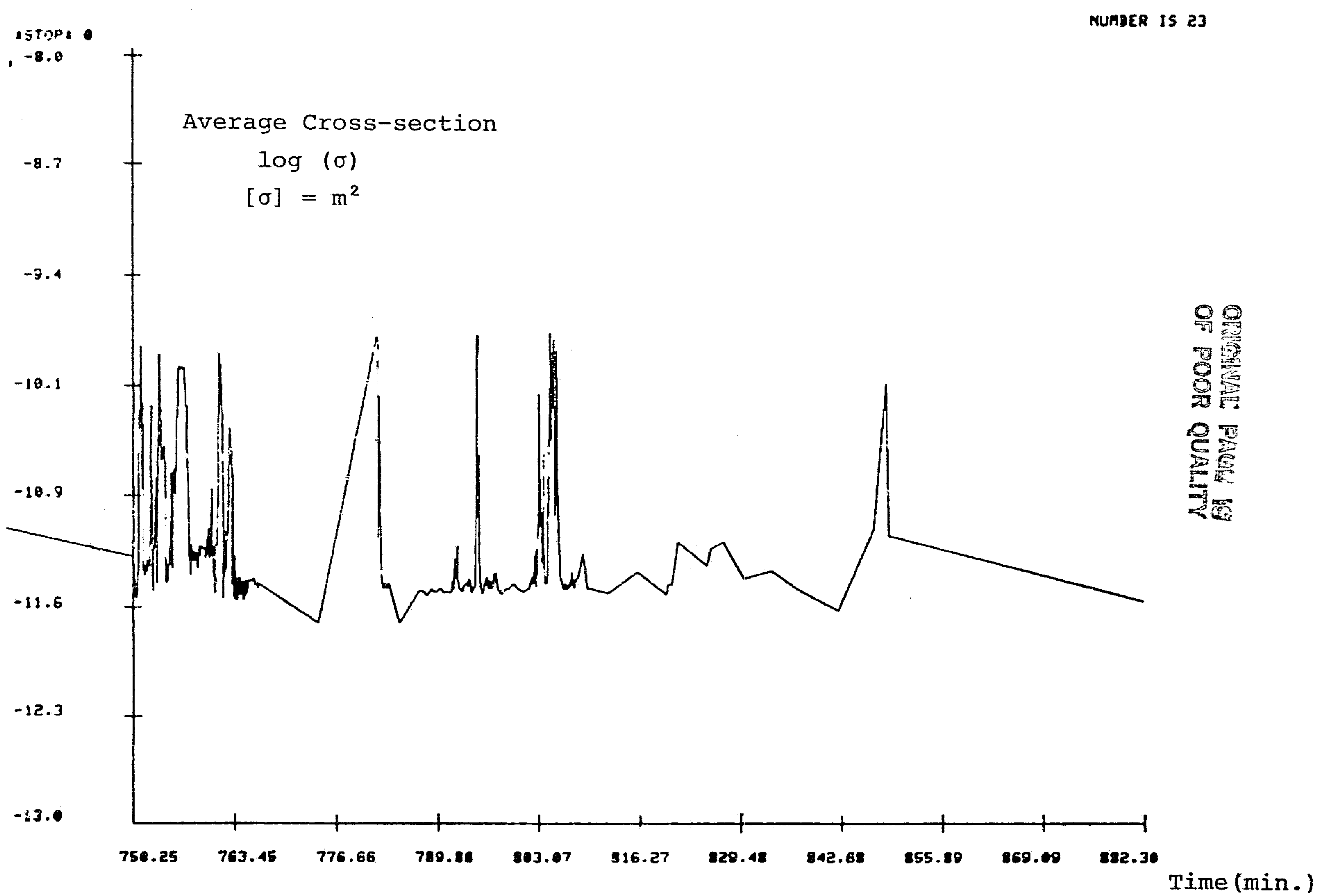


Fig. 49 Average Cross-section σ

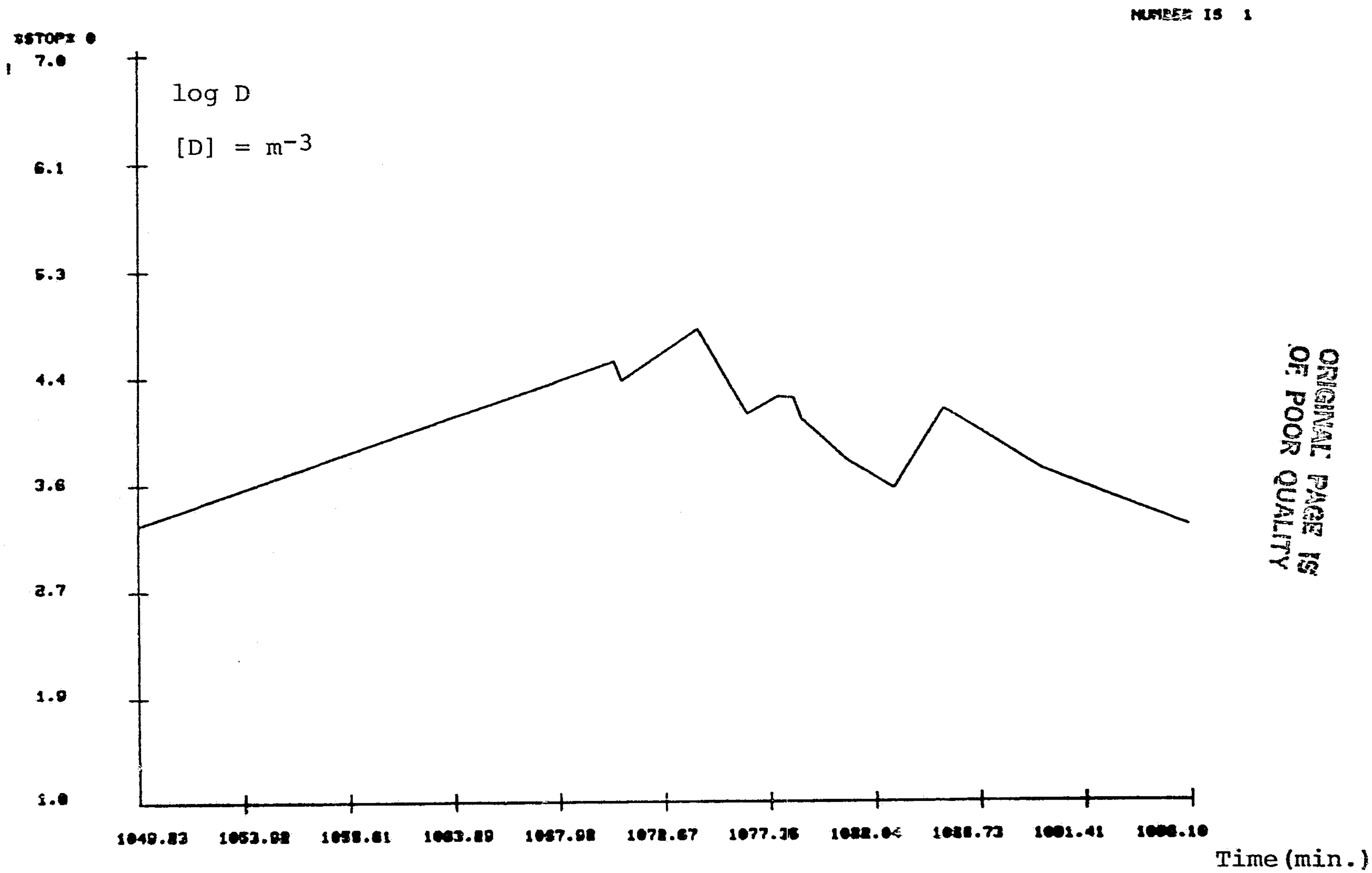


Fig. 50 Particle Density D

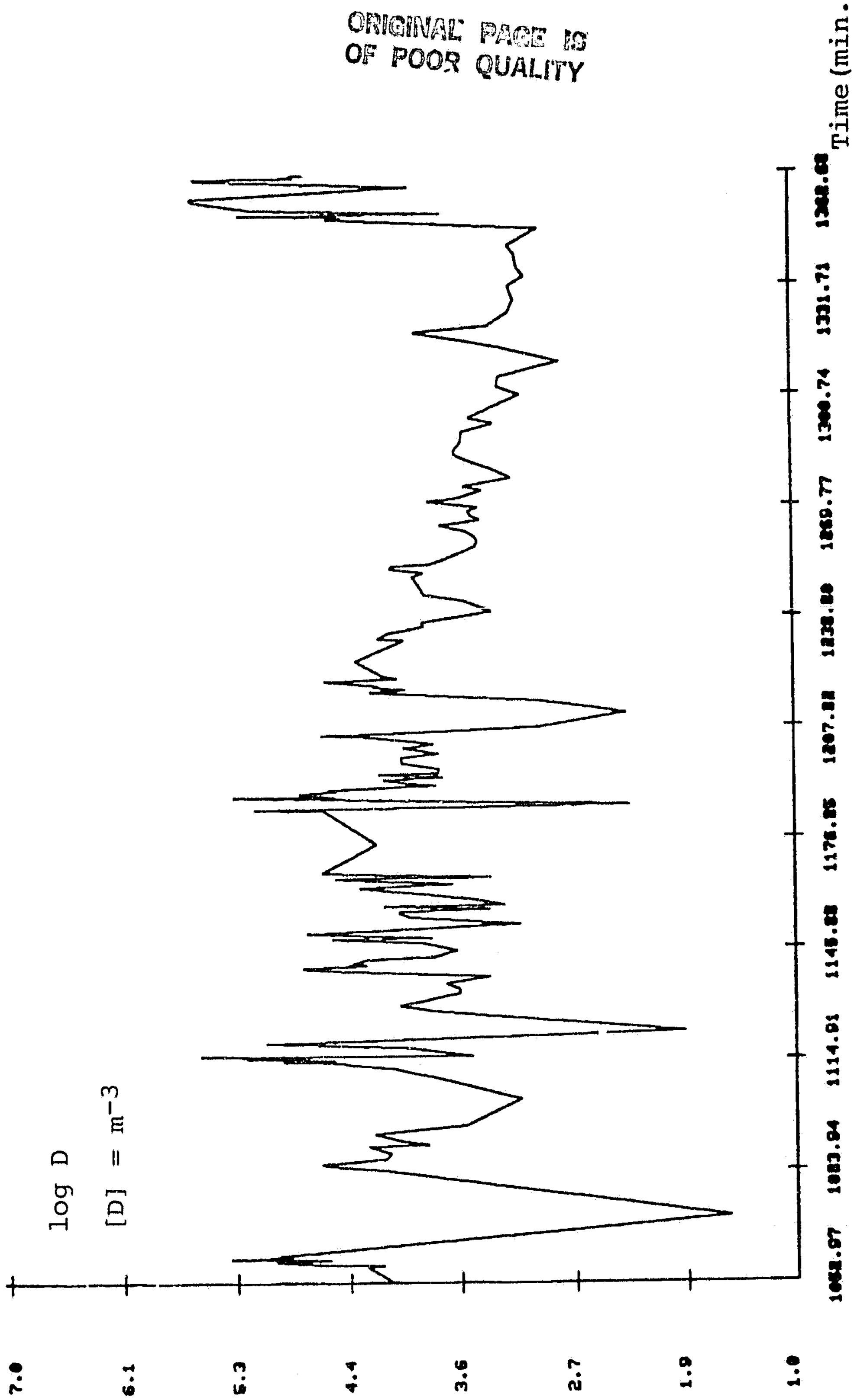


Fig. 51 Particle Density D

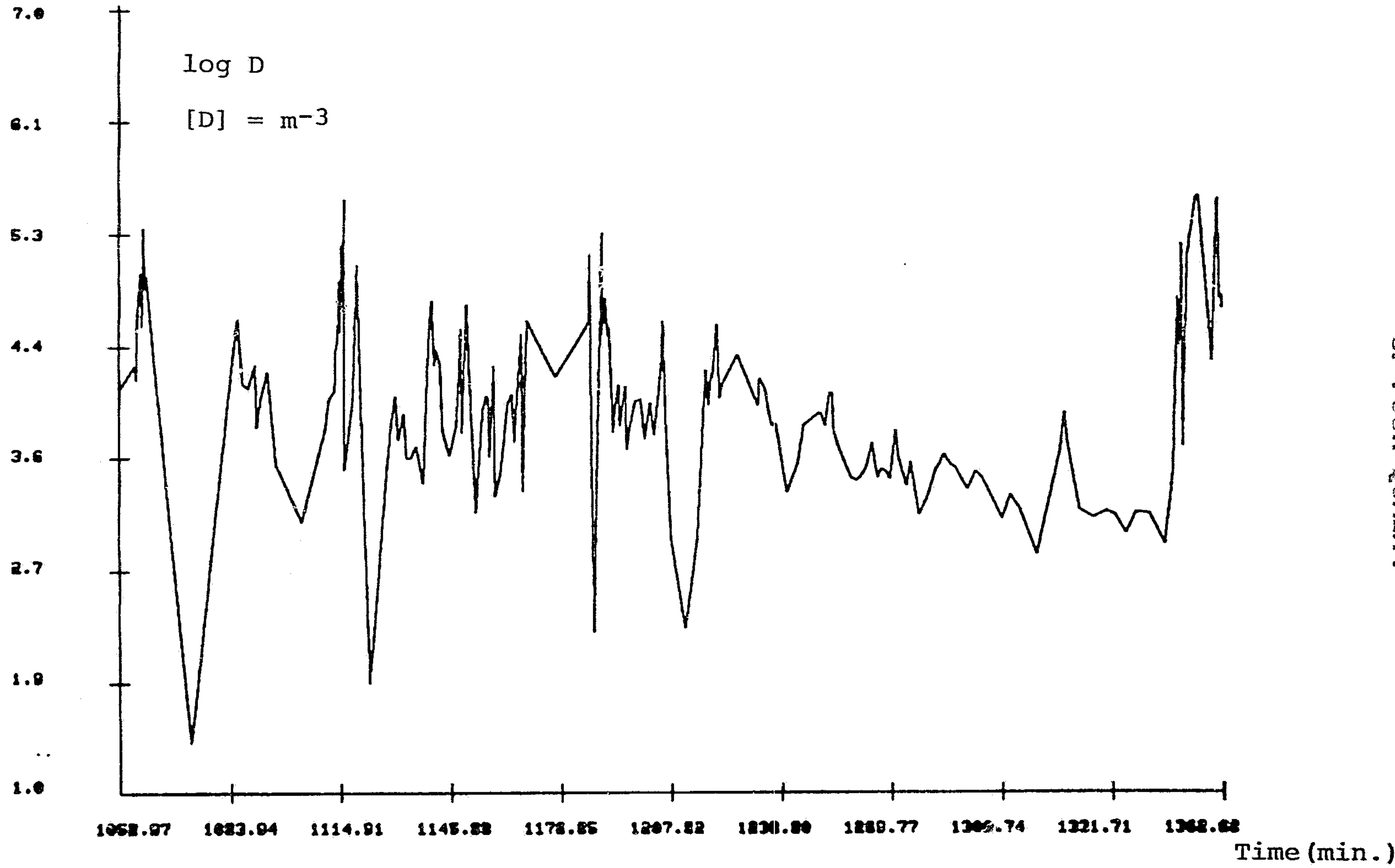


Fig. 52 Particle Density D

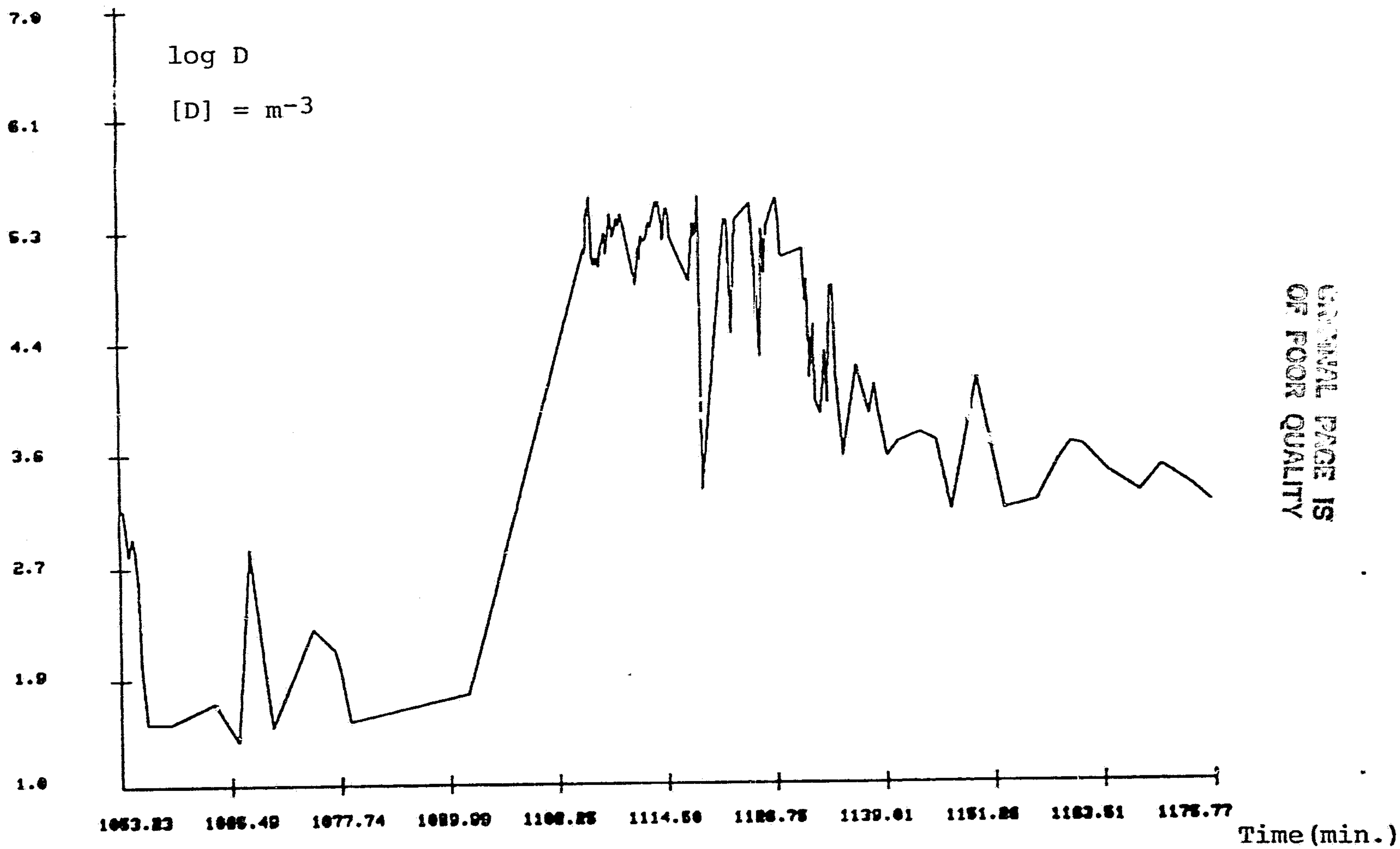


Fig. 53 Particle Density D

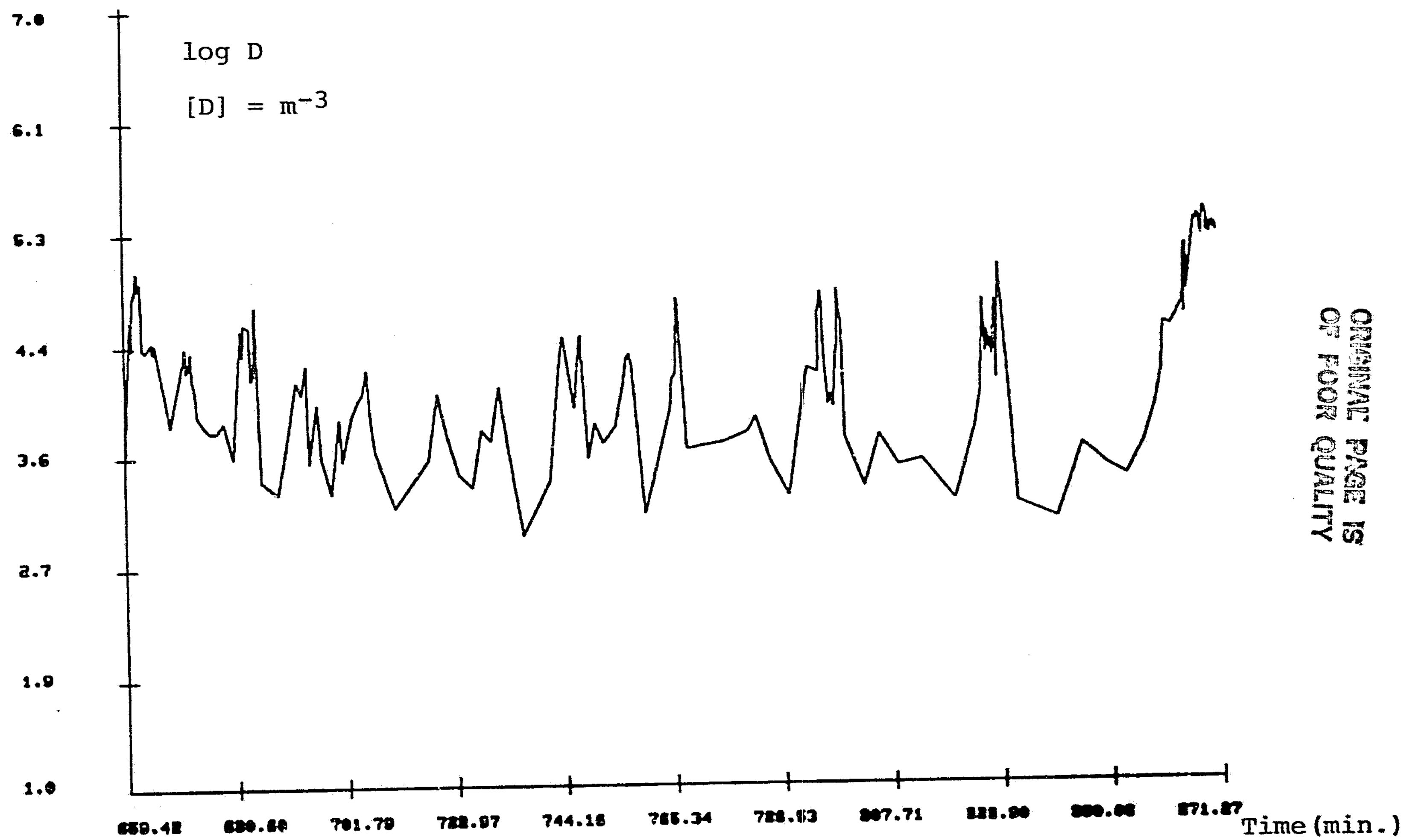


Fig. 54 Particle Density D

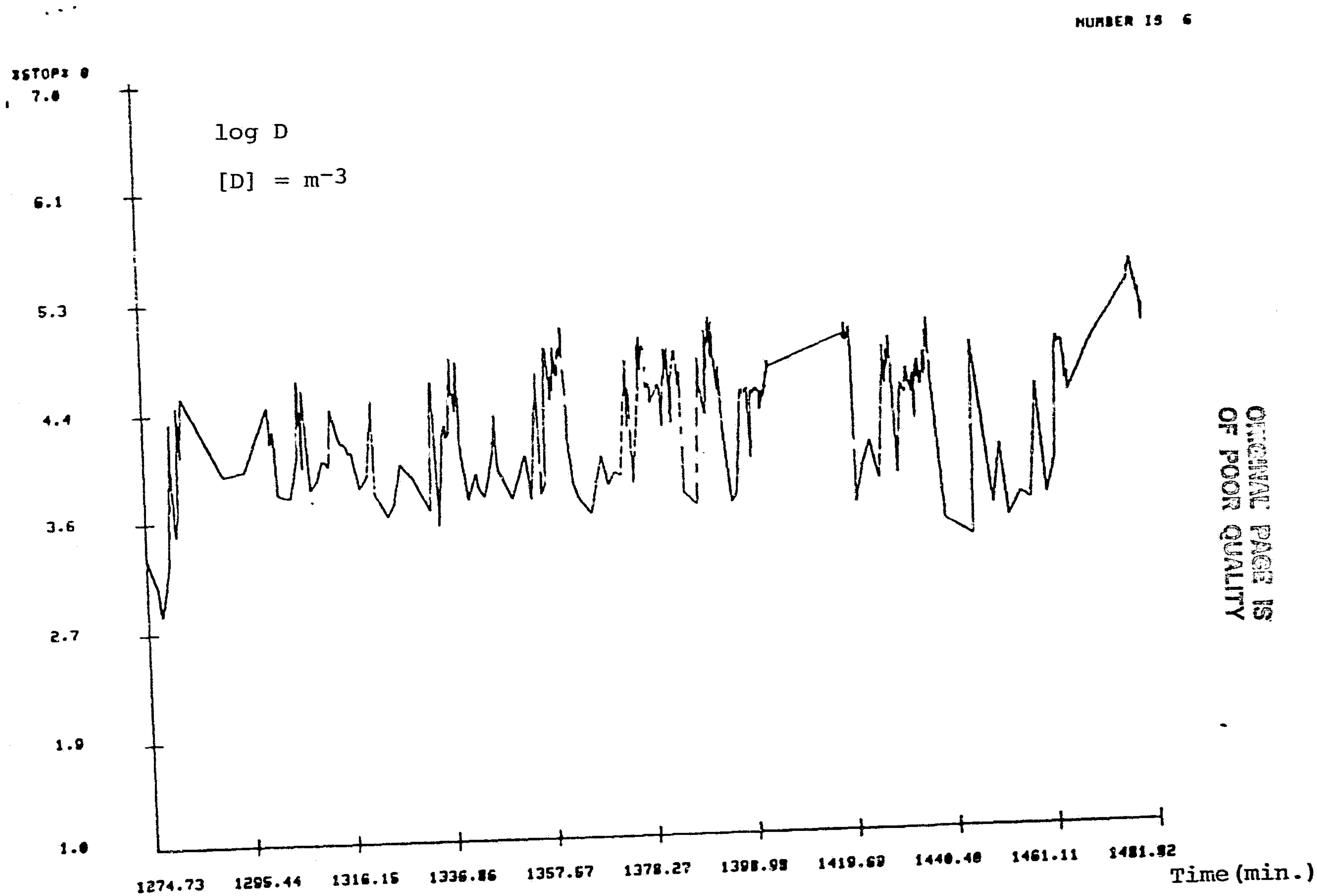


Fig. 55 Particle Density D

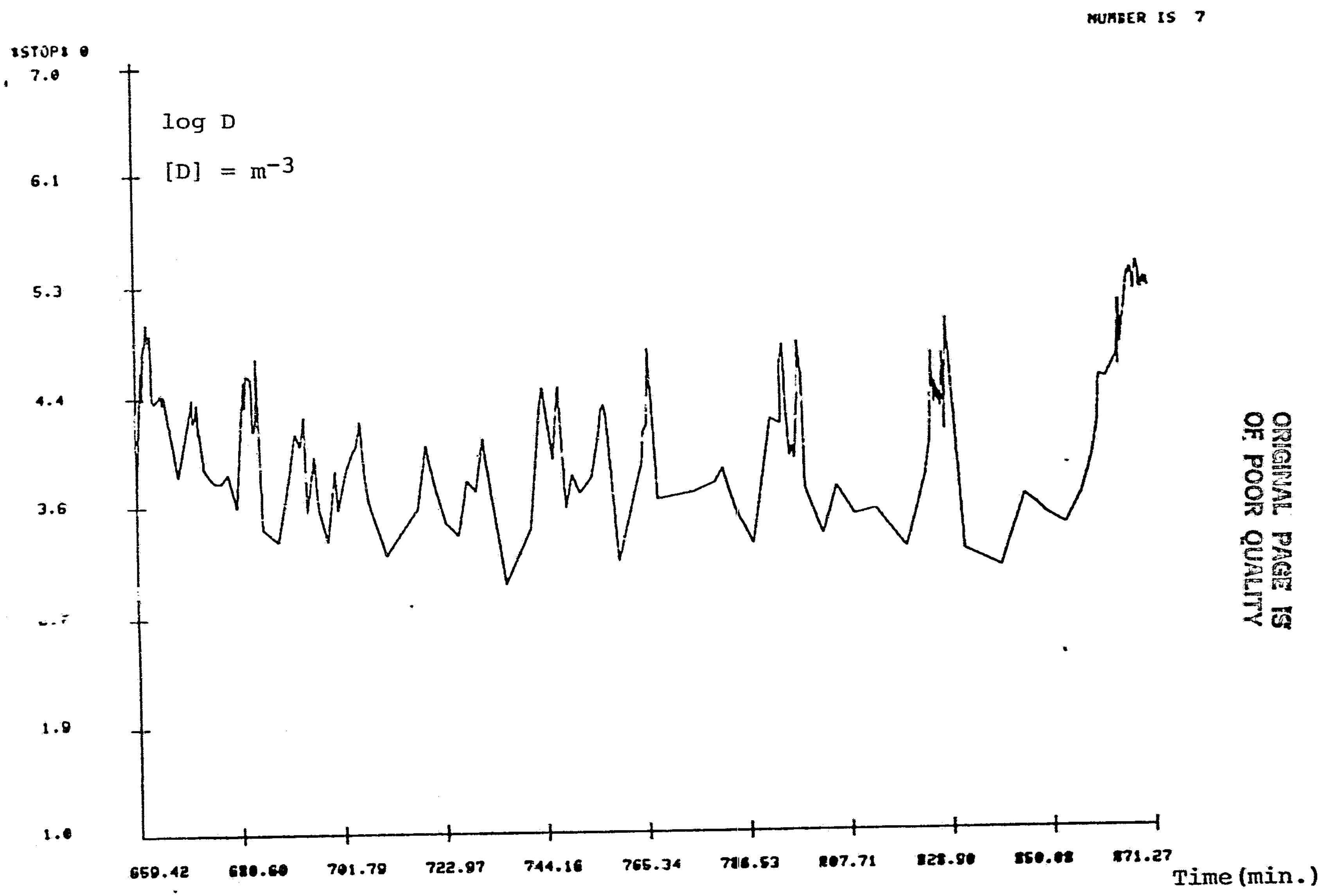


Fig. 56 Particle Density D

ORIGINAL PAGE IS
OF POOR QUALITY

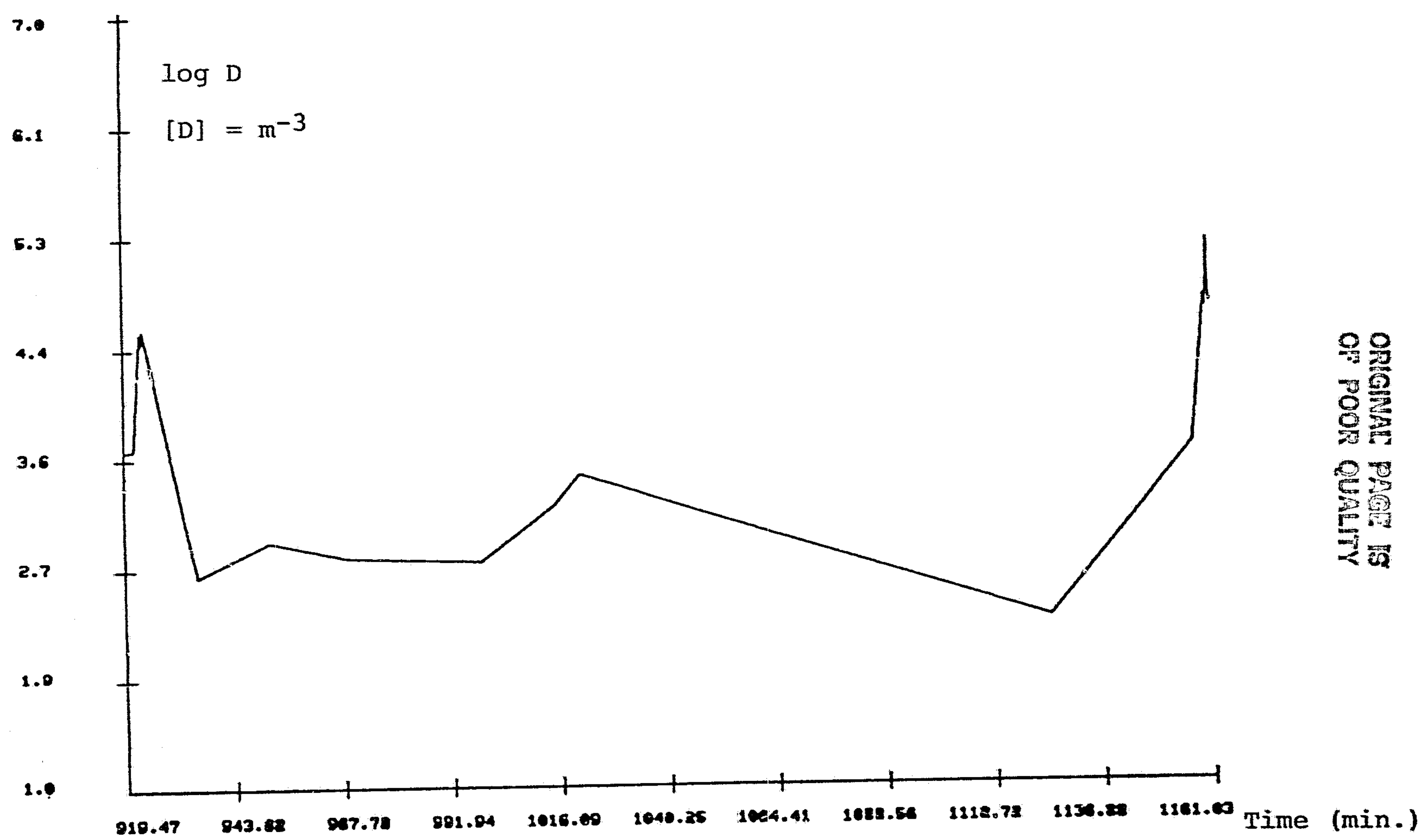


Fig. 57 Particle Density D

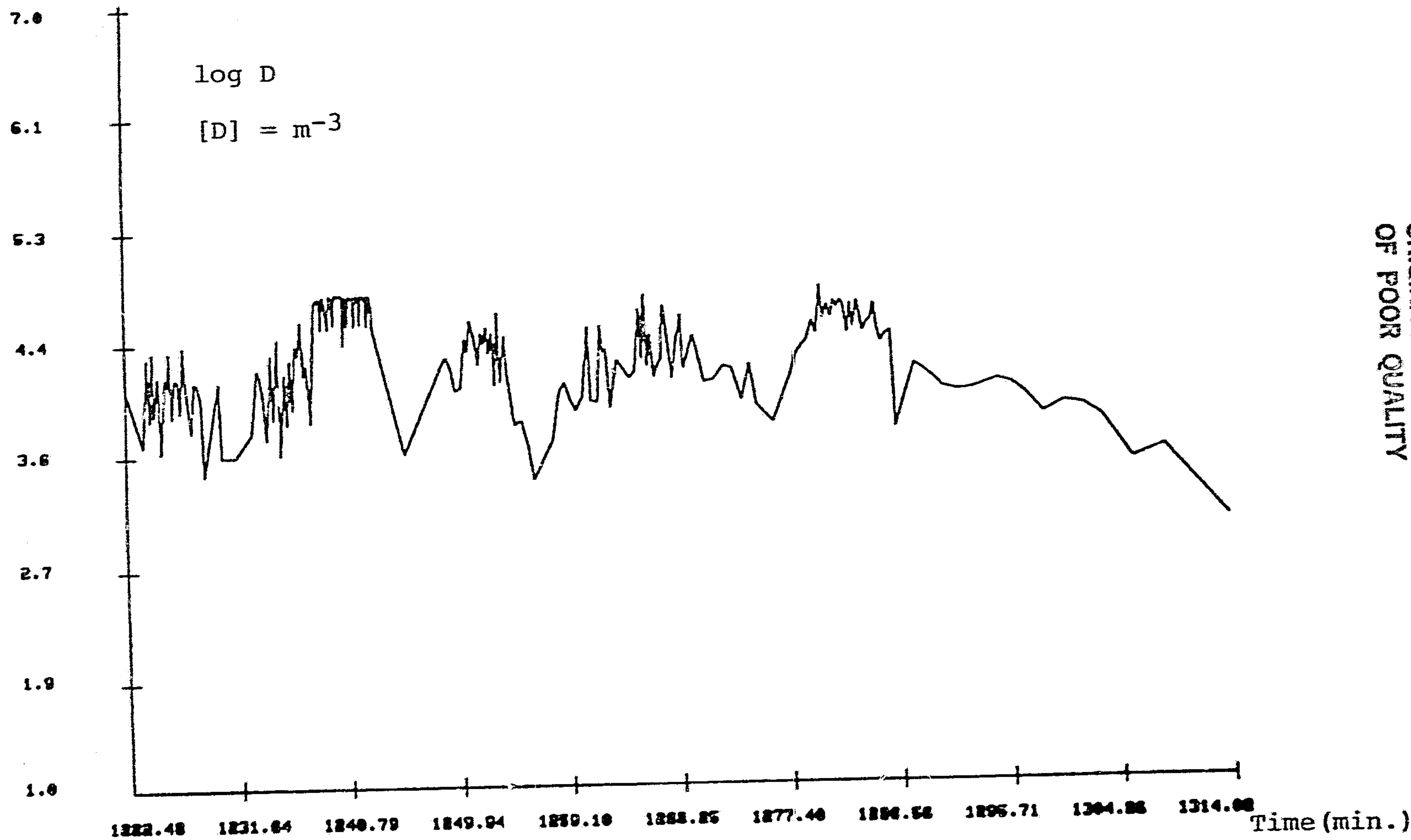


Fig. 58 Particle Density D

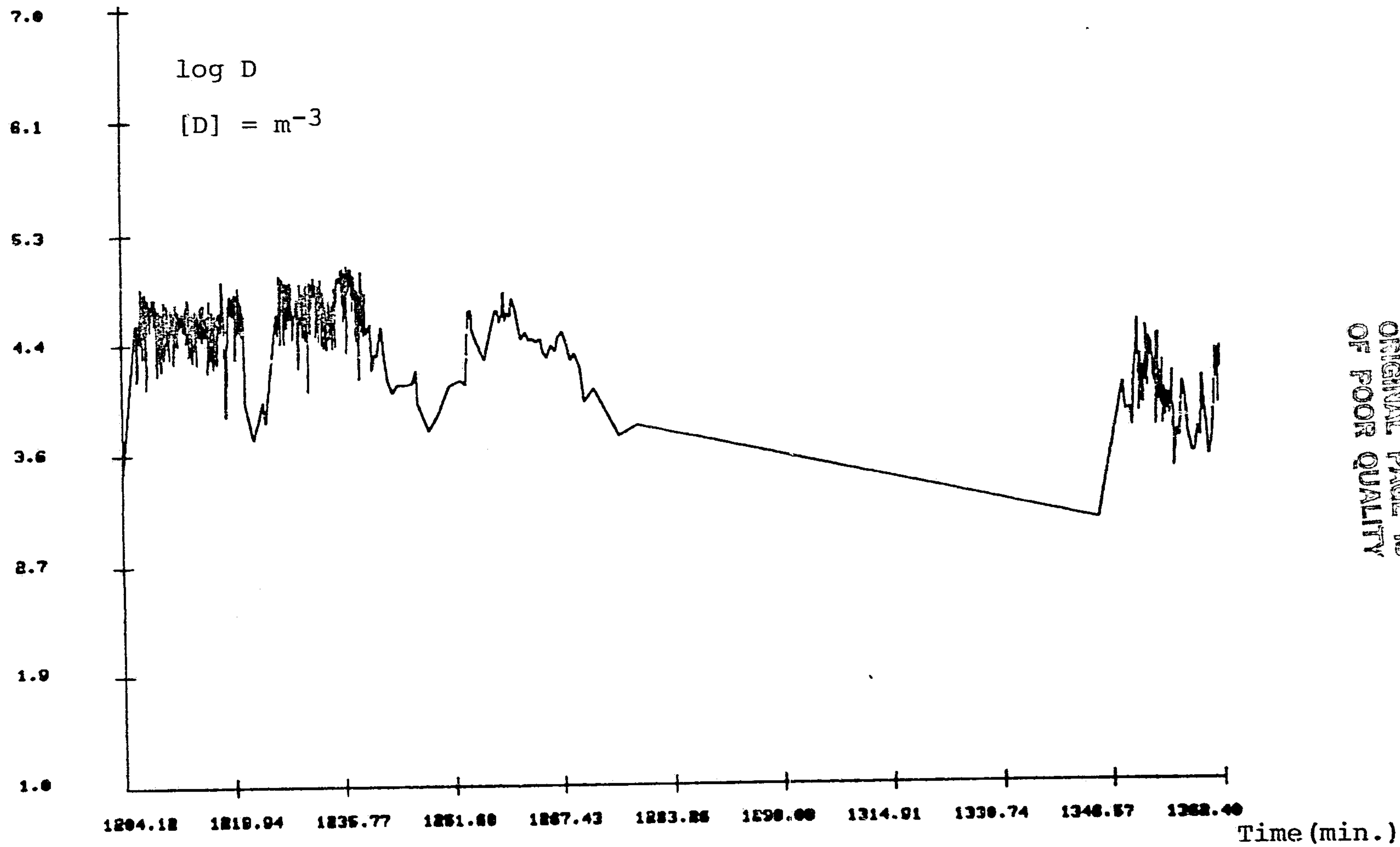
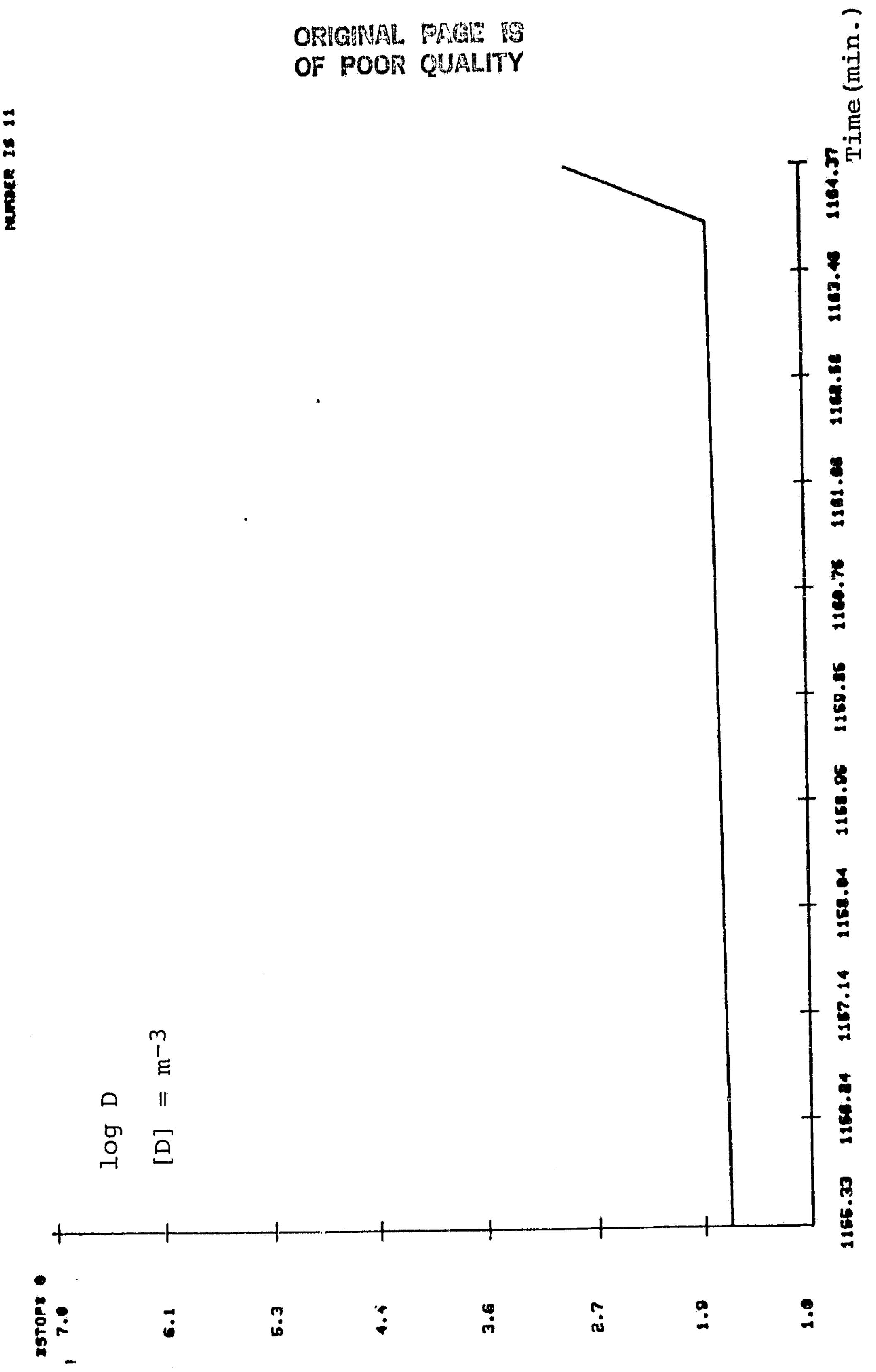


Fig. 59 Particle Density D

ORIGINAL PAGE IS
OF POOR QUALITY



Particle Density D

Fig. 60

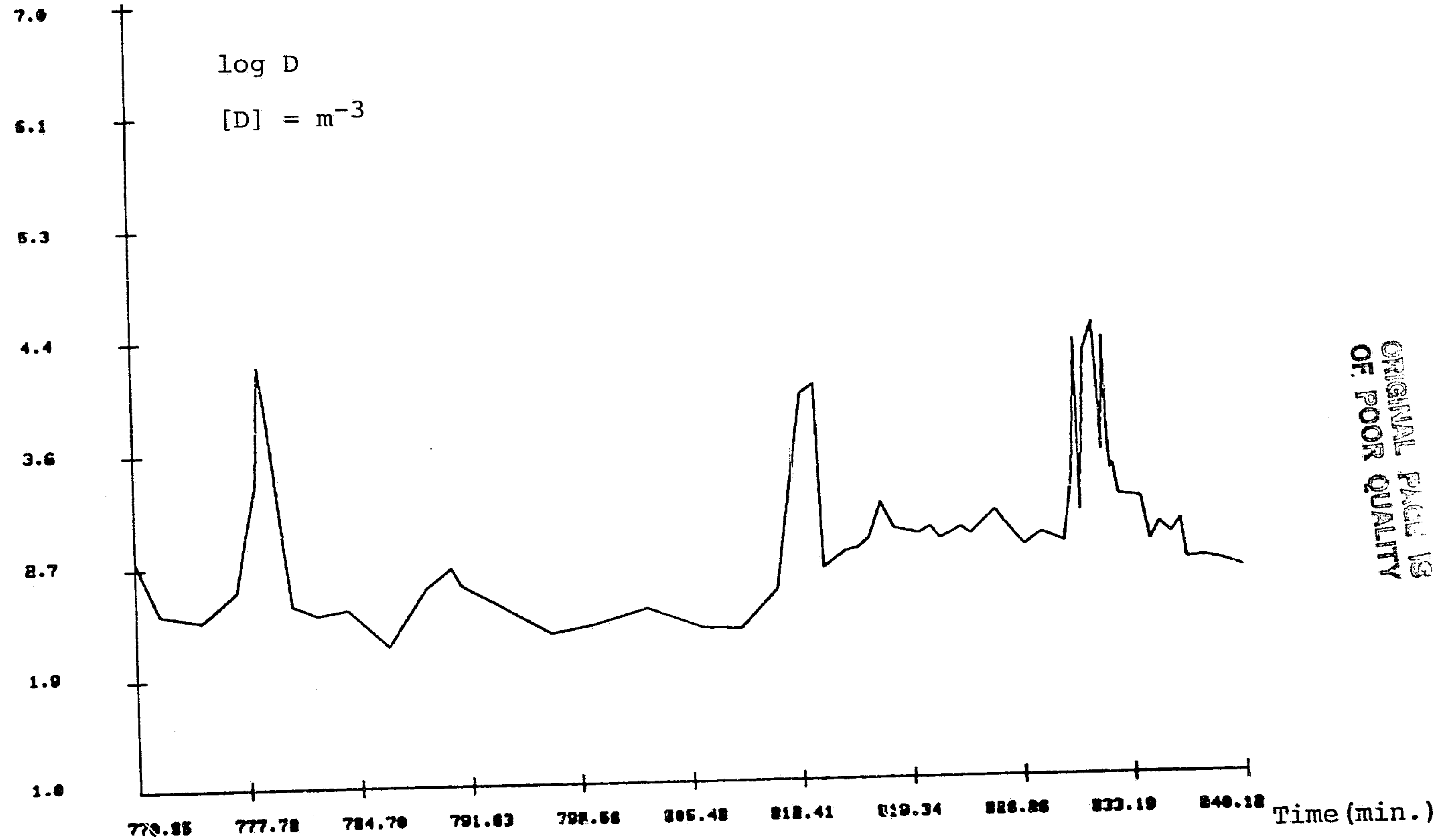


Fig. 61   Particle Density D

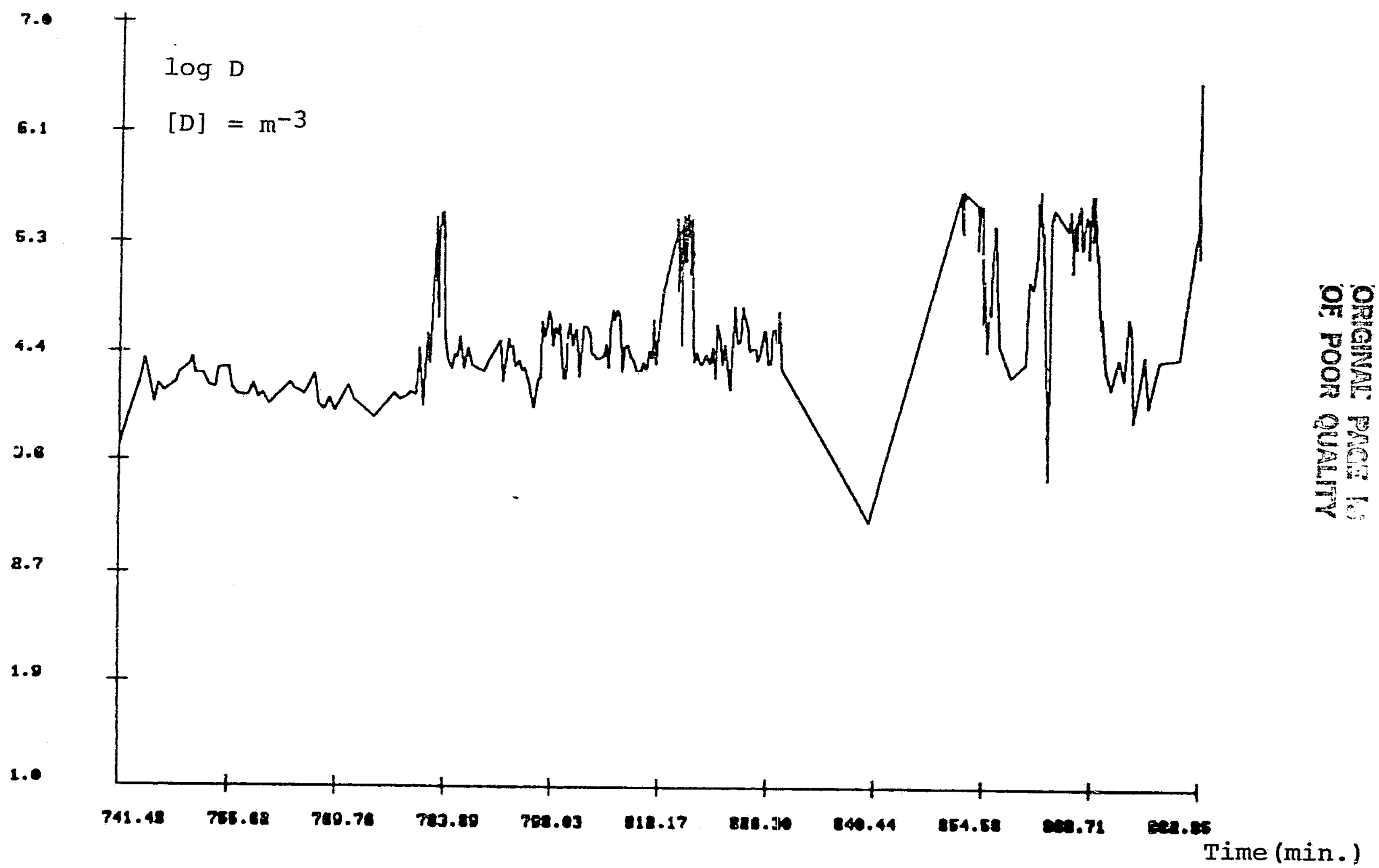


Fig. 62 Particle Density D

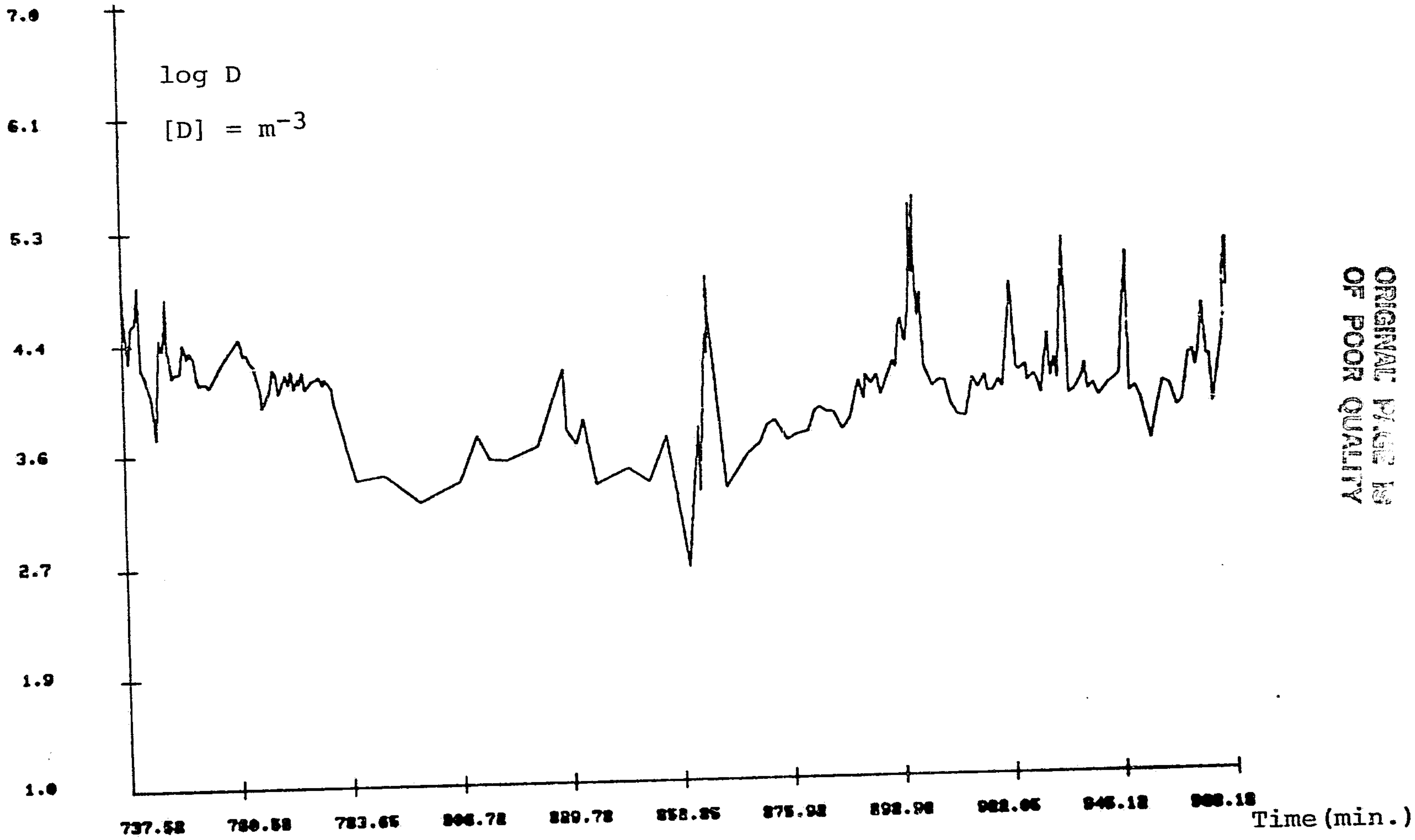


Fig. 63 Particle Density D

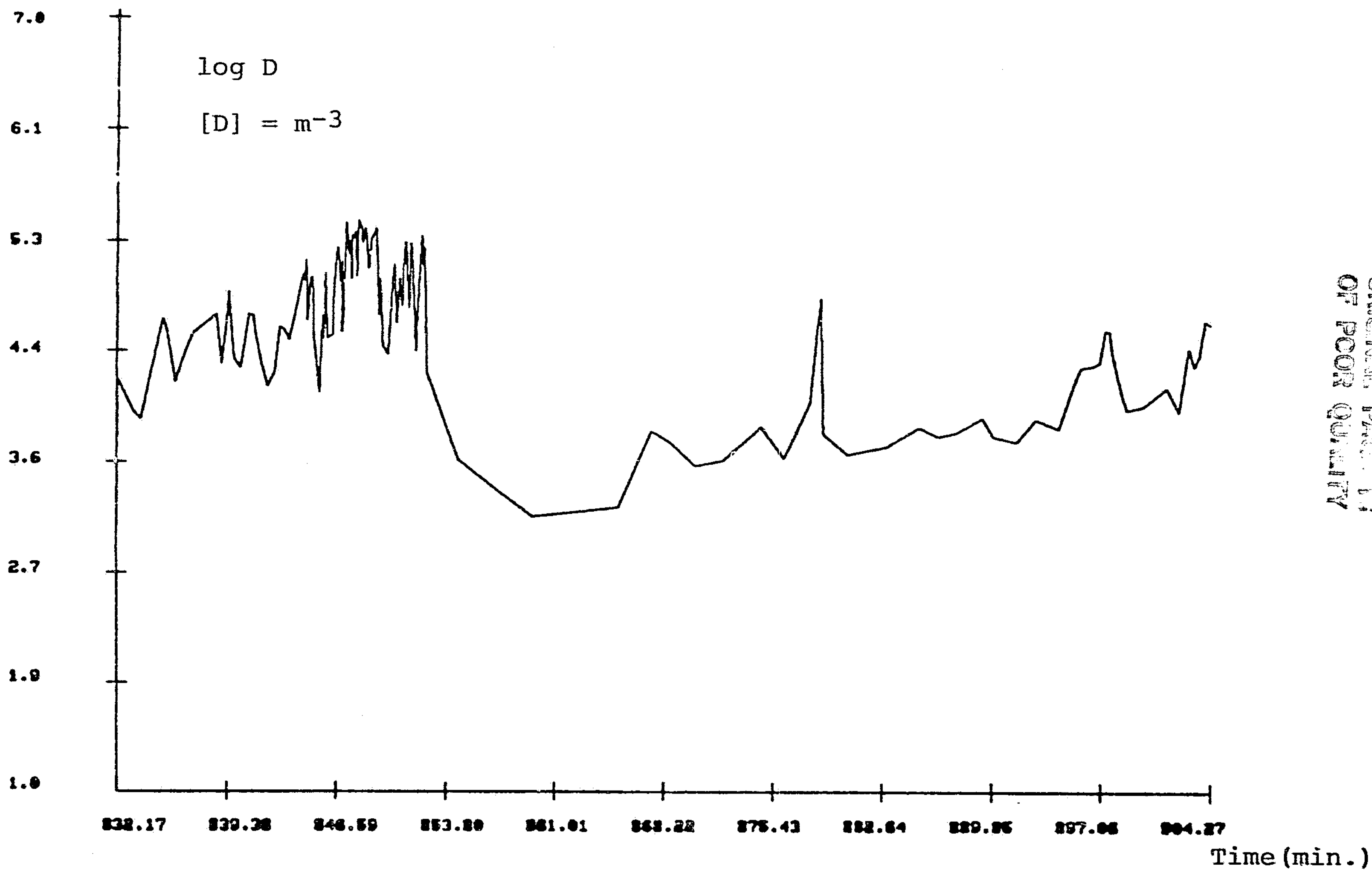


Fig. 64 Particle Density D

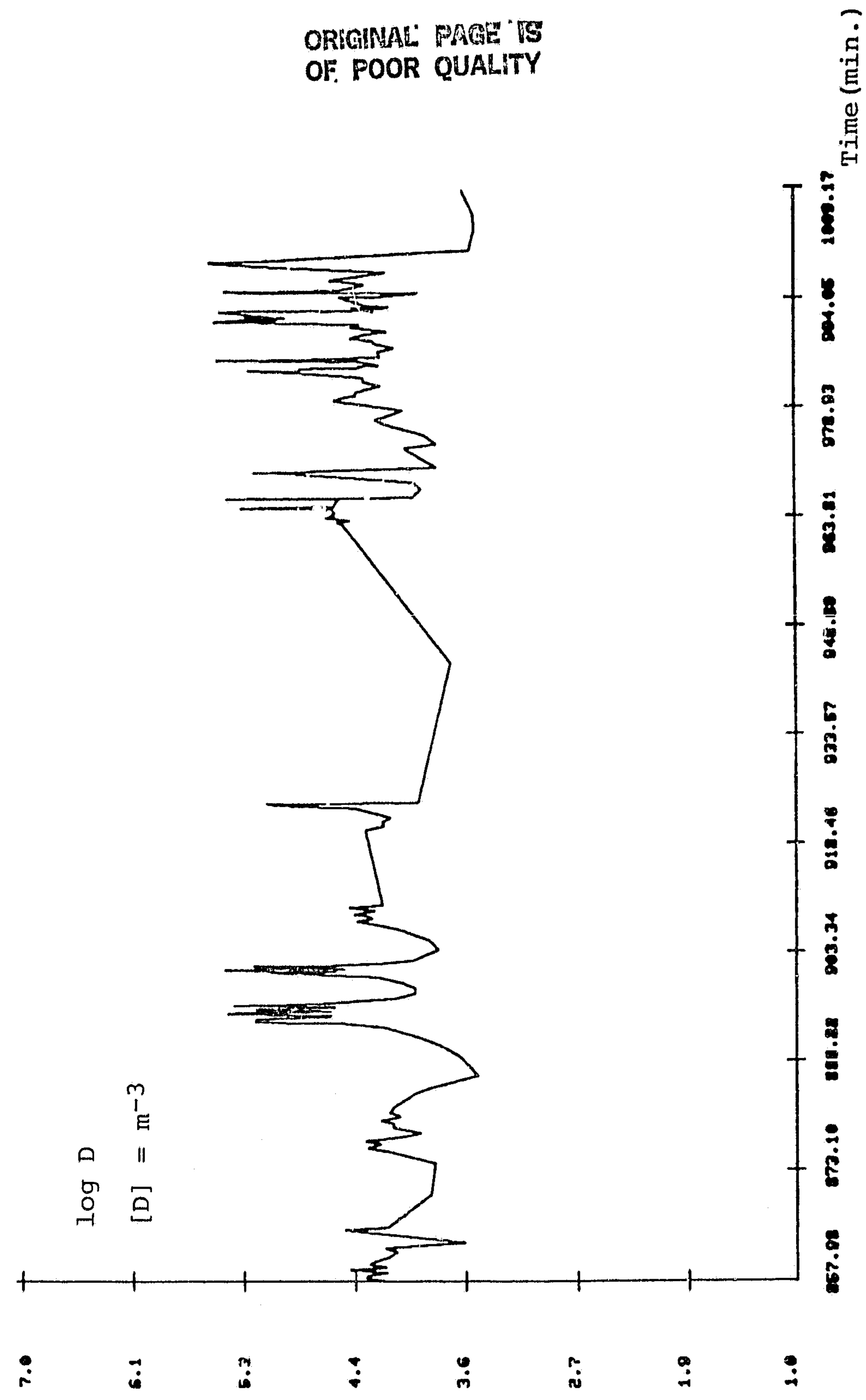


Fig. 65 Particle Density D

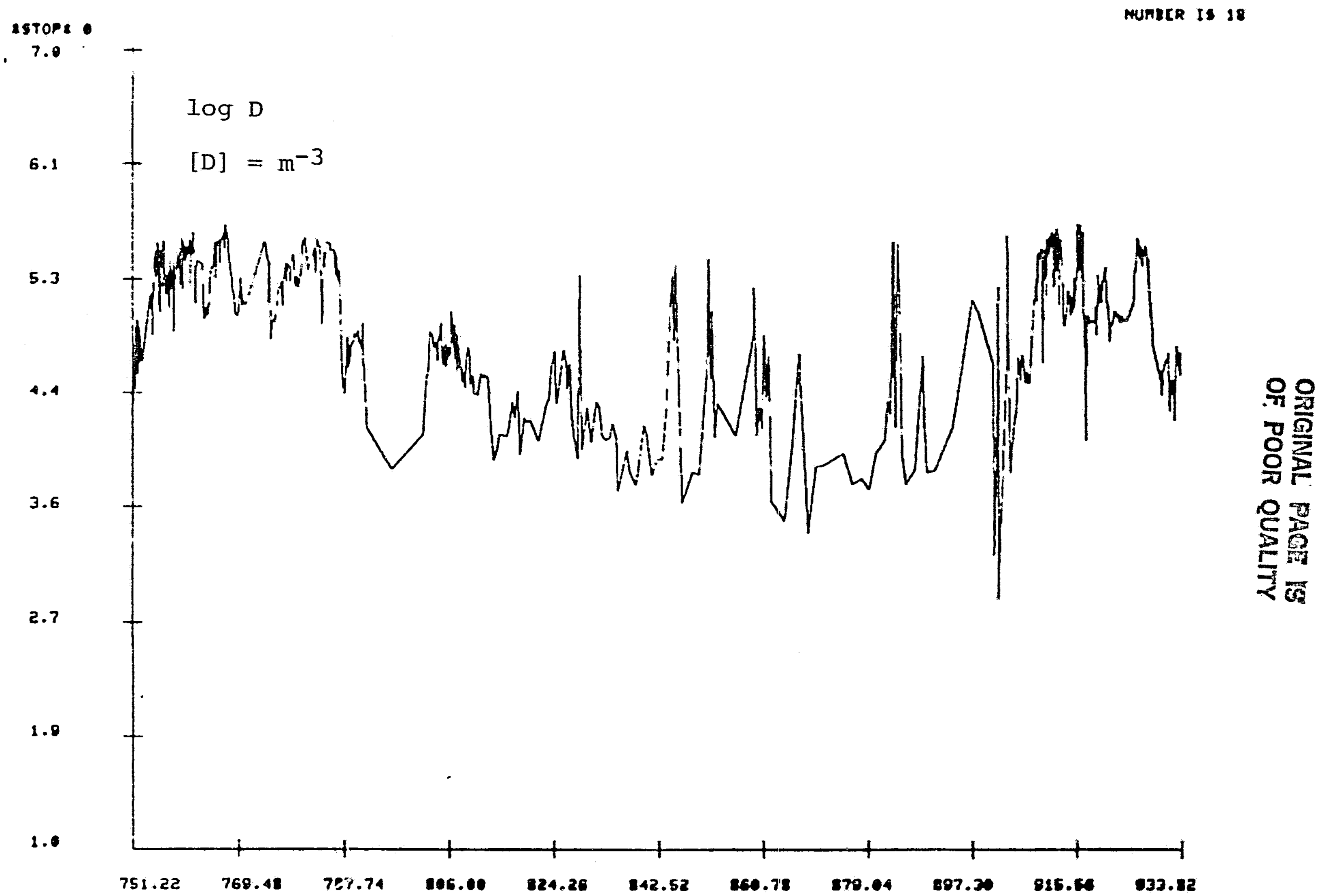


Fig. 66 Particle Density D

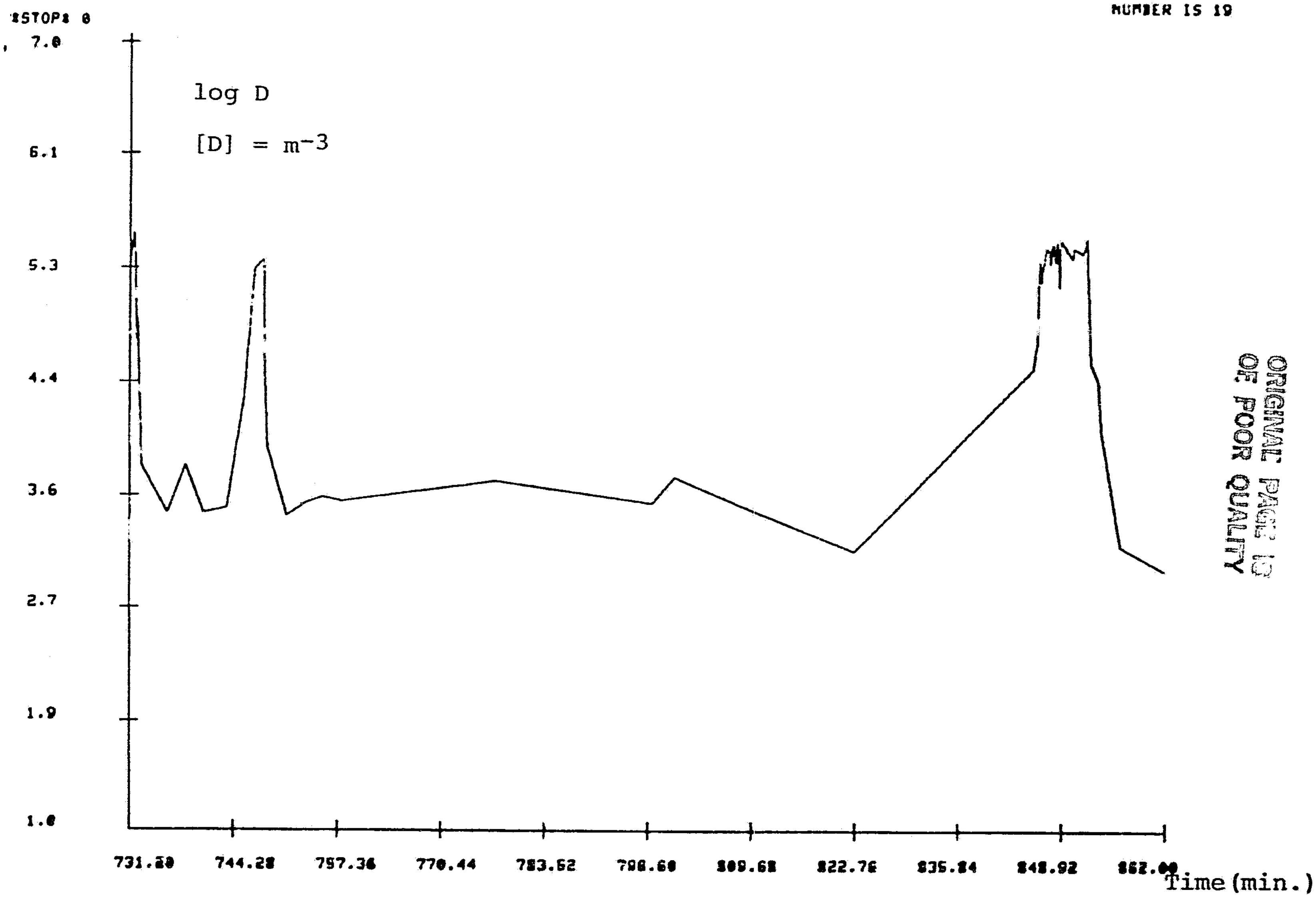


Fig. 67 Particle Density D

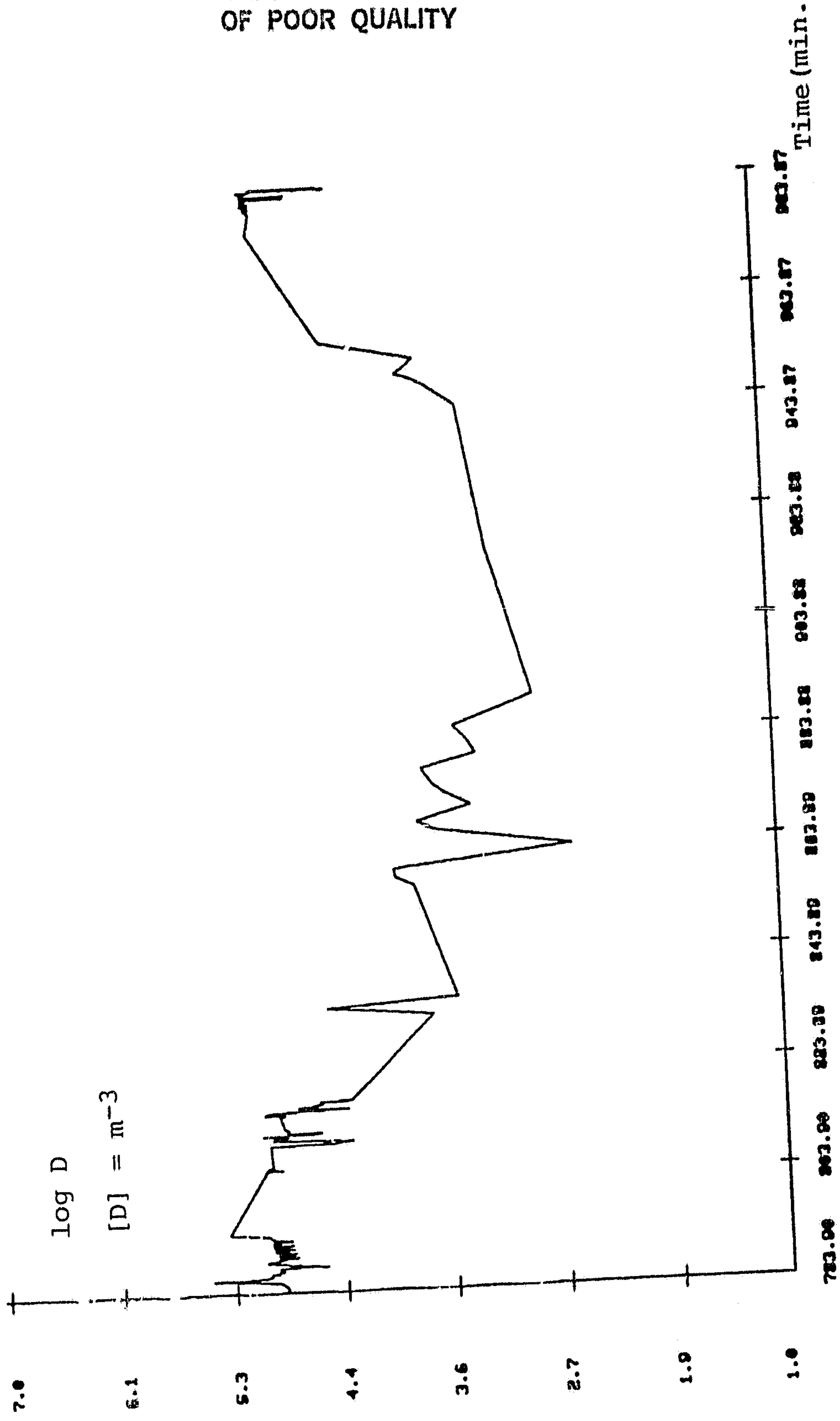


Fig. 68 Particle Density D

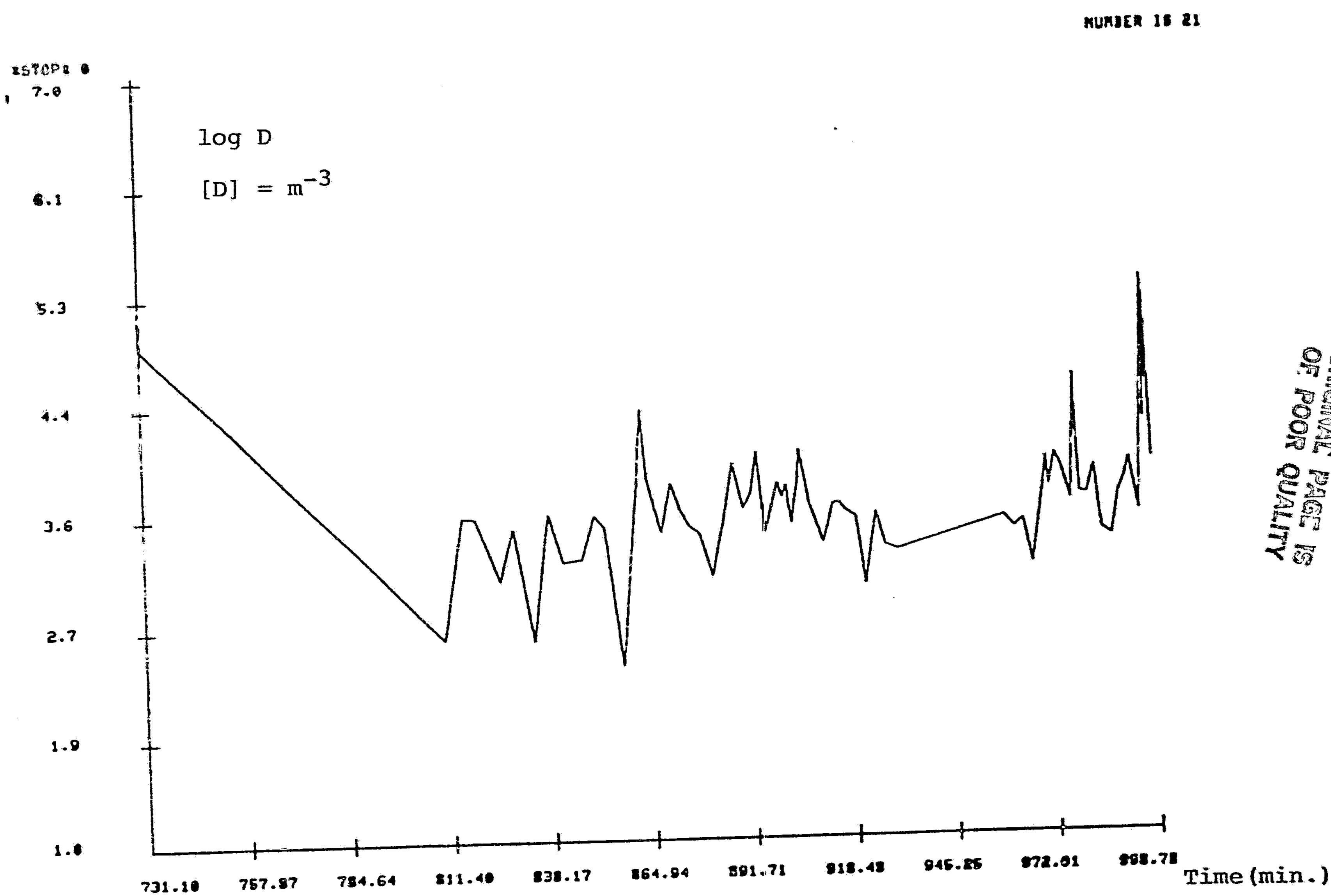


Fig. 69 Particle Density D

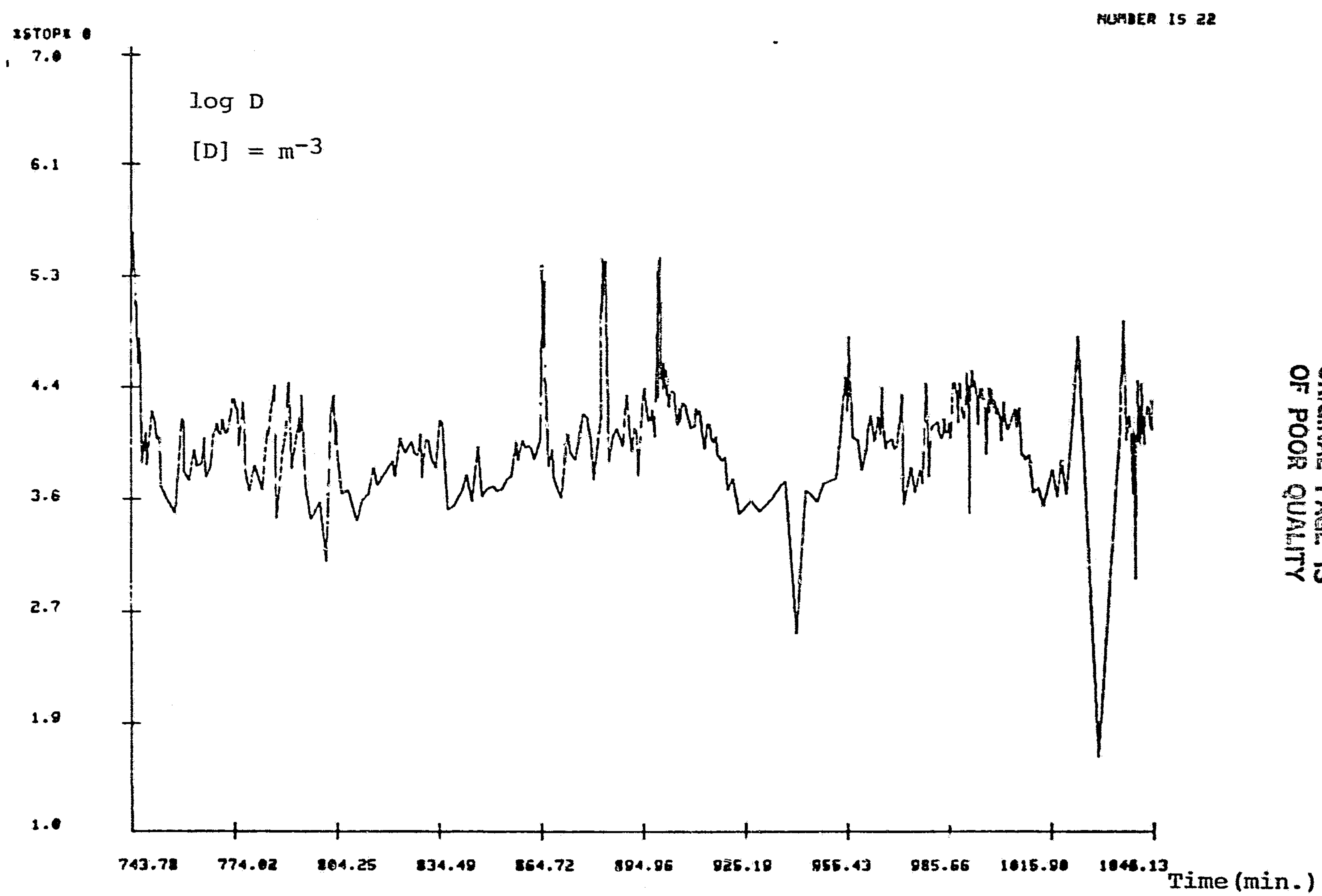


Fig. 70 Particle Density D

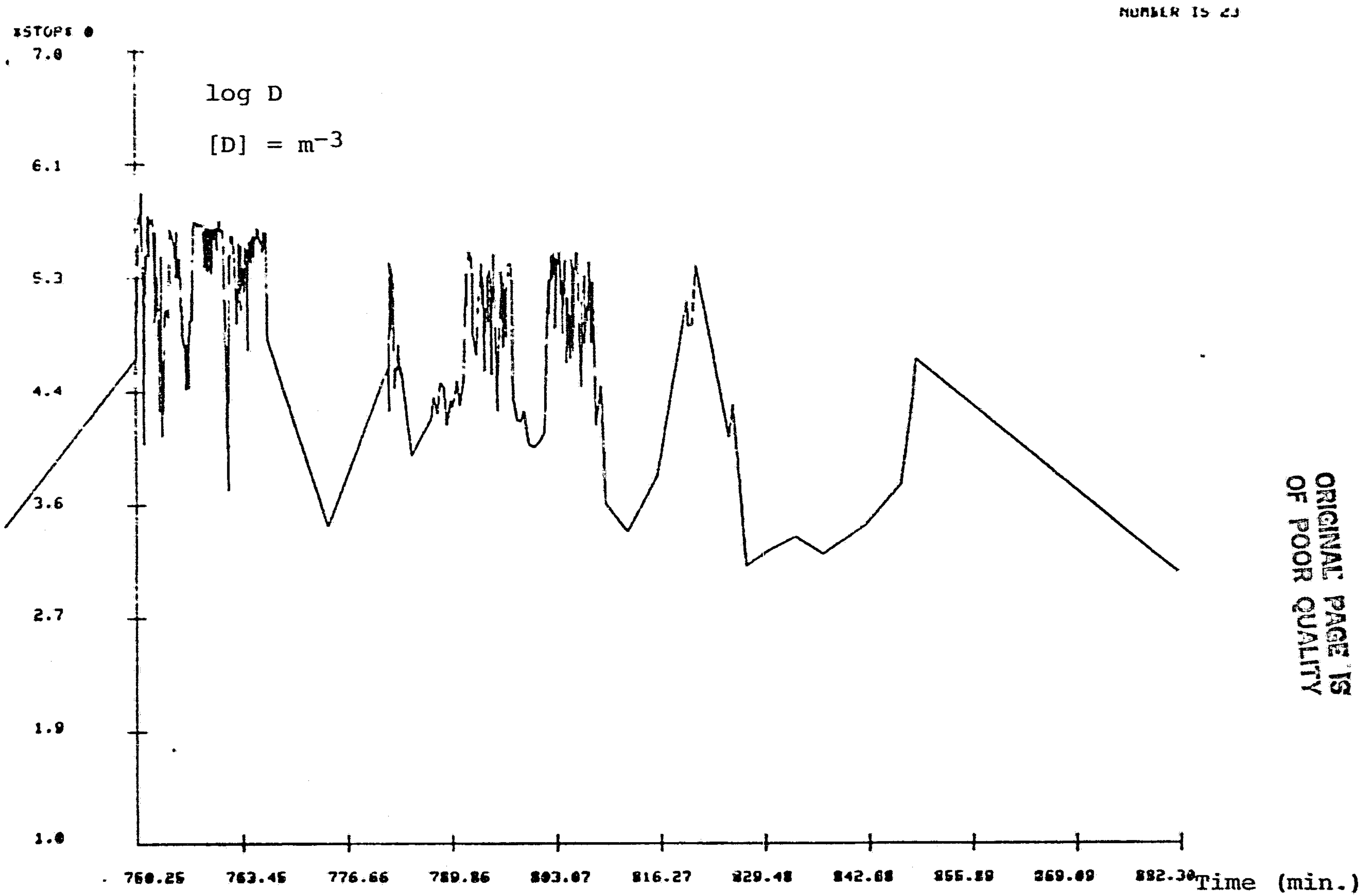


Fig. 71 Particle Density D

3. LDV SIGNAL PROCESSING ANALYSIS AND SIMULATION

Applied Research has utilized and modified a signal processing code written by the Wave Propagation Laboratory of NOAA. This code was written for a Data General machine and utilized on the Sigma V computer at MSFC. Conversion and implementation of this code constituted a significant task. A listing of this code as it was utilized in this study is given in Appendix B. A flow chart of the complete code is given in Figure 72.

Basically the code is capable of generating a sinusoidal signal with noise or a narrow band random process signal with noise, and processing this signal according to selected algorithms to give frequency estimates. It is the narrow band signal which is appropriate for LDV studies. The code was used in these studies to generate the standard deviation of the mean frequency estimate as a function of the input signal-to-noise for a narrow band random process, over an ensemble of 1000 different cases for each point. This quantity leads directly to the expected velocity error for a given case.

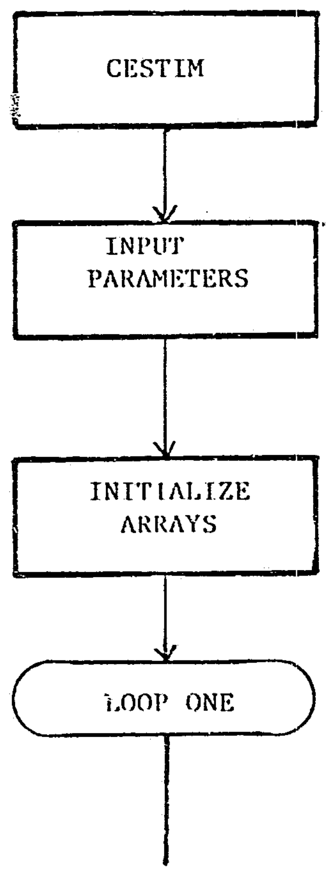
Use of the Code

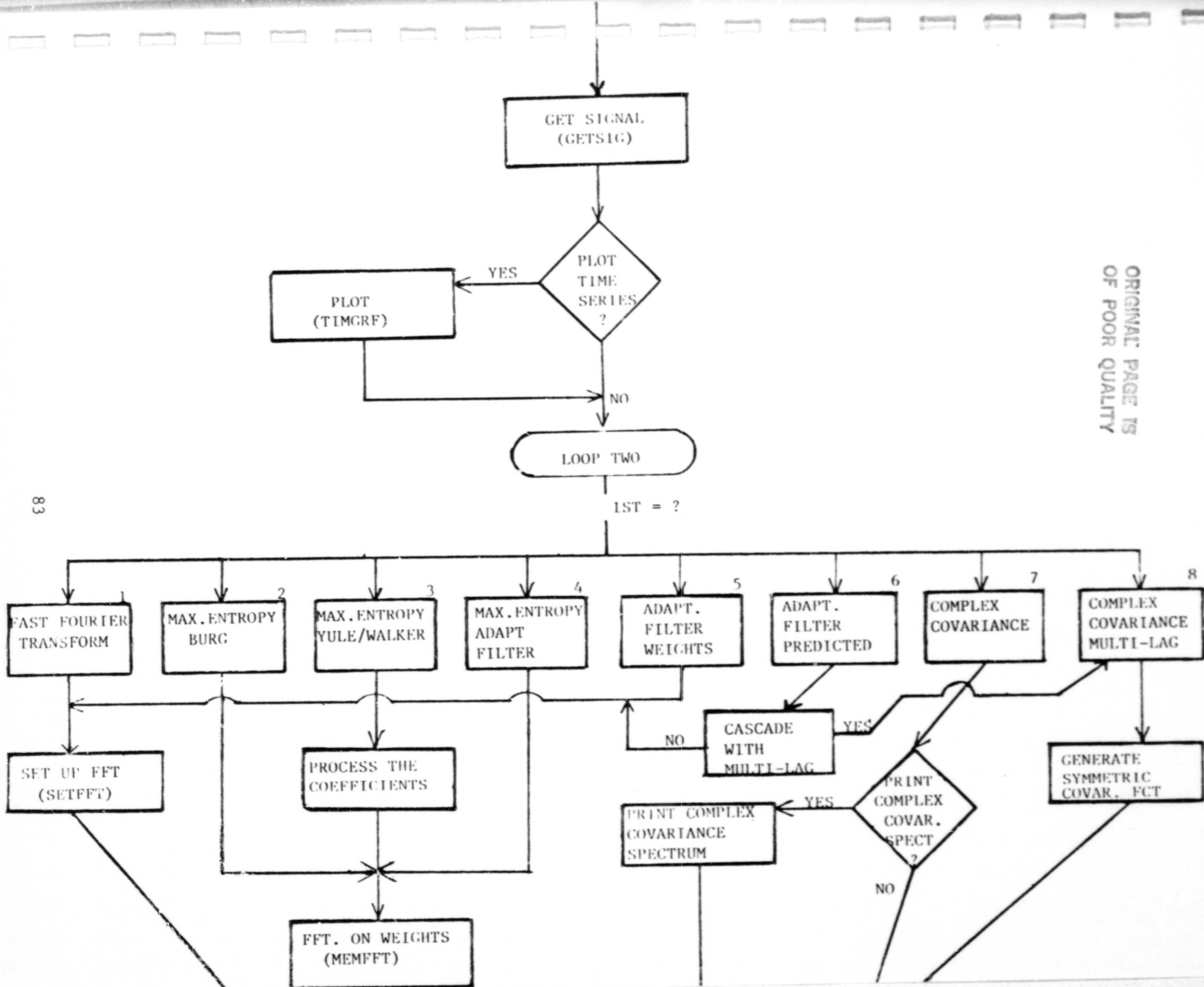
The signal processing program used by Applied Research, Inc. is listed on the NASA computer system as ARIDAN and XARIDAN. The ARIDAN is the fortran listing of the program, while XARIDAN is the executable source.

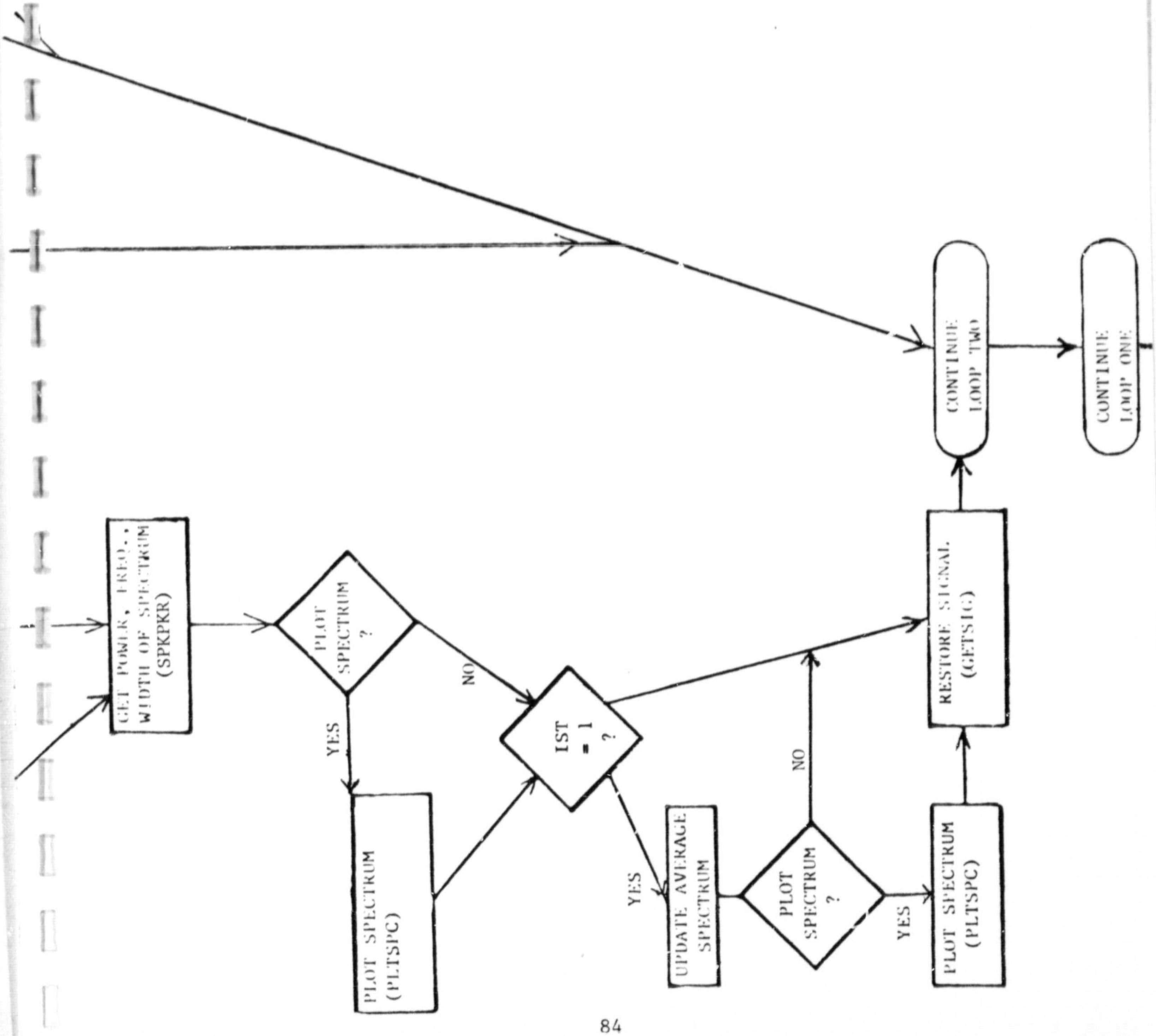
There are two ways of running the program. One is to execute a command file labeled DANCMD. This will set the logical unit number ten for terminal output. The second way is for the user to set the logical unit ten for a terminal output, then start XARIDAN.

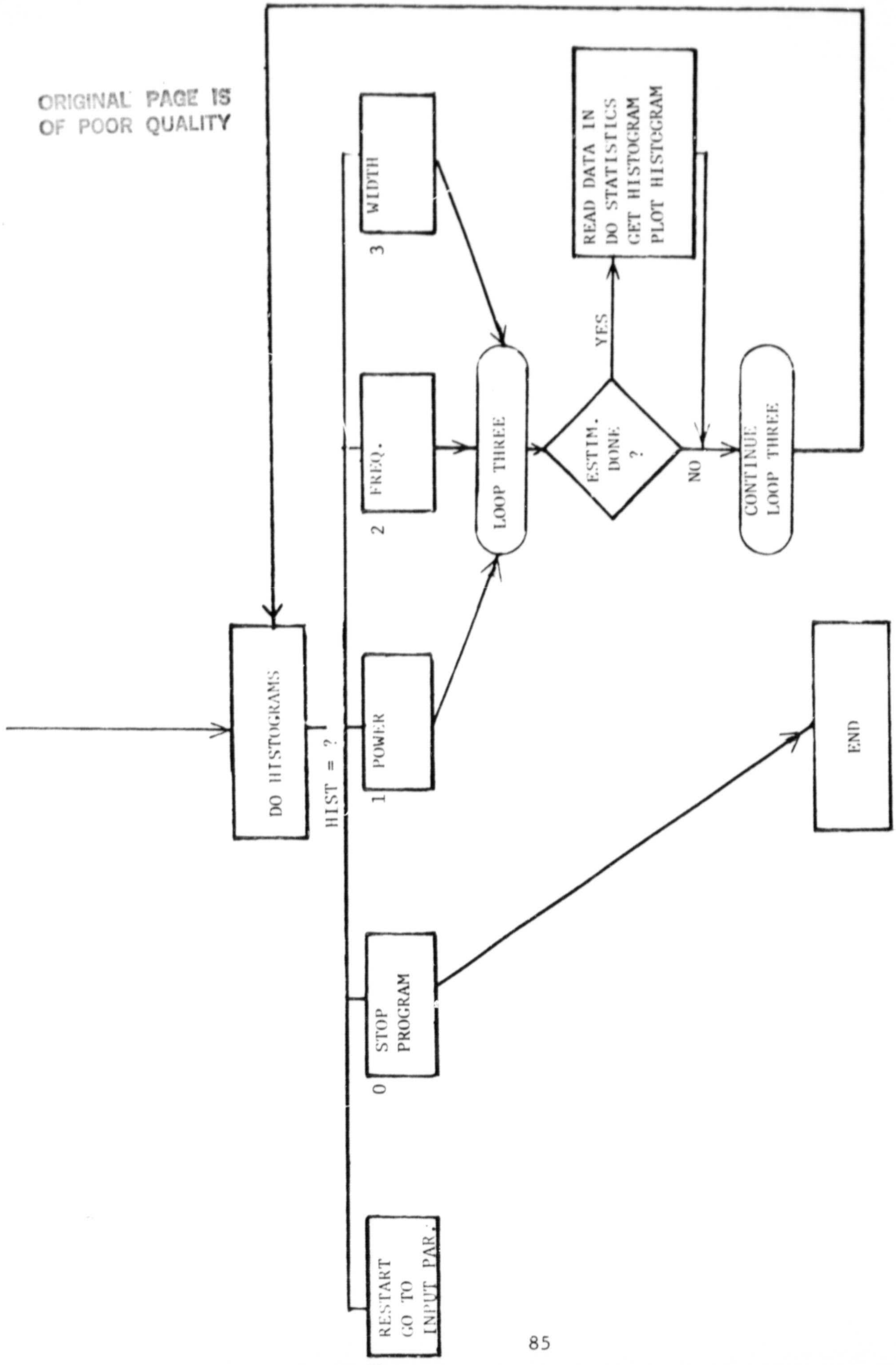
Once the program has been started, data input will be requested. The

Fig. 72 CESTIM FLOW CHART




GET SIGNAL (GETSIG)
PLOT TIME SERIES ?
YES
NO
PLOT (TIMGRF)
LOOP TWO
IST = ?
1
2
3
4
5
6
7
8
FAST FOURIER TRANSFORM
MAX.ENTROPY BURG
MAX.ENTROPY YULE/WALKER
MAX.ENTROPY ADAPT FILTER
ADAPT. FILTER WEIGHTS
ADAPT. FILTER PREDICTED
COMPLEX COVARIANCE
COMPLEX COVARIANCE MULTI-LAG
CASCADE WITH MULTI-LAG
YES
NO
SET UP FFT (SETFFT)
PROCESS THE COEFFICIENTS
FFT. ON WEIGHTS (MEMFFT)
PRINT COMPLEX COVARIANCE SPECTRUM
PRINT COMPLEX COVAR. SPECT ?
YES
NO
GENERATE SYMMETRIC COVAR. FCT


GET POWER, FREQ., WIDTH OF SPECTRUM (SPKPKR)
PLOT SPECTRUM ?
YES
NO
PLOT SPECTRUM (PLTSPC)
IST = 1 ?
YES
NO
UPDATE AVERAGE SPECTRUM
PLOT SPECTRUM ?
YES
NO
PLOT SPECTRUM (PLTSPC)
RESTORE SIGNAL (GETSIG)
CONTINUE LOOP TWO
CONTINUE LOOP ONE


DO HISTOGRAMS
HIST = ?
-1
RESTART
GO TO
INPUT PAR.
0
STOP
PROGRAM
1
POWER
2
FREQ.
3
WIDTH
LOOP THREE
ESTIM.
DONE
?
YES
NO
READ DATA IN
DO STATISTICS
GET HISTOGRAM
PLOT HISTOGRAM
CONTINUE
LOOP THREE
END

program will need to know if the user wants sinusoidal or narrow band random process synthetic data. The program will then list the available estimators and give the user an option to use or not use them. Then depending on which estimators are being used, the program will ask for inputs needed to run those estimators.

Inputs that are common to every estimator are: (1) the frequency of synthetic data (FREQ); (2) for narrow band random processes, the standard deviation of the signal (SIGMA); (3) the signal to noise ratio of data (SNR); (4) the number of input data points (NTOT); (5) the length of the transform for spectrum (NPTS); (6) Hanning window for fourier transform (IWNDW); (7) the number of estimates (NEST); (8) type of spectrum plot (LOGSCL); and (9) does user want a hard copy of input parameters.

The number of input data points and the length of the transform determine the size of the signal generated. If NTOT is less than NPTS the program generates a signal of size NPTS. If NTOT is greater than NPTS the program will generate a signal with the size the closest power of two larger than NTOT.

When the Fast Fourier Transform estimator is used the system will ask for the FFT averaging time constant. This parameter is used to set the number of times a block average will be done.

When the Adaptive Filter - Weights estimator is used the system will need to know the order of the filter (LORD) and also the normalized adaptive constant (ALPHA).

When Adaptive Filter - Predicted estimator is used the system will need the same parameters as Adaptive Filter - Weights but will also require some others. The program will need to know the number of samples to use in the prediction filter. It will also want to know if user wants to cascade the

filter with the Multi-Lag Complex Covariance estimator. This option was added by Applied Research.

Finally, when the Multi-Lag Complex Covariance estimator is used the system will need to know the order of the estimator.

After inputs, the program will clear the screen and make the plot of the first time series and ask user if a hard copy is wanted.

The program will then proceed to the signal processing and a plot of the power spectrum for each estimator will be made. After the processing is done the terminal bell will ring twice and a question mark will be displayed. This gives the user the time to make a hard copy from the terminal and then press return to continue.

The program then clears display and asks the user if a histogram is wanted. Choices of histograms are power, frequency, and width. If user does not want a histogram, the user may either restart program without exiting, or end processing.

Analysis and Simulation of Processing Methods

In a previous report (Final Report on Contract No. NAS8-34337, June 1982) the application of adaptive filtering to LDV processing has been considered. It was seen (Figure 6.5) that an adaptive linear predictive filter (ALPF) offers several dB improvement over the pulse pair processor within a large range of S/N values for the input signal. However, the theoretical limit (Cramer-Rao Bound) for improvement is still beyond this, indicating that significant further inprovement is possible. The poly-pulse-pair (PPP) processor was know to follow the linear predictive filter result but theoretical or simulated results for the ALPF preceding the PPP processor had not been obtained. Under the present task, results for the PPP processor (on the figure called multi-lag L=4 case), and the ALPF have been separately simulated. Then the two processors have been cascaded with the ALPF preceding the PPP. As a baseline, the result of an FFT on the input has been calculated using the multi-lag L=1 algorithm.

The processing simulated was taken to be appropriate to the MSFC pulsed system, with a sampling rate of 15 MHz. A Doppler frequency of 4.5 MHz was assumed, although this is incidental to the result. A narrow band random process signal was synthetically generated with a standard deviation of .02 yielding a velocity standard deviation of .48 m/sec for a total spread of about 1 m/sec. A data sequence of 64 points was used, corresponding to a 4 microsecond pulse with a length of 1200 meters.

The PPP processor was taken to have four lags, and the ALPF was taken to have four weights. The convergence factor, α, was taken to be .1 (See R. J. Keeler, NOAA Technical Memorandum ERL WPL-49, p. 15).

ORIGINAL PAGE IS
OF POOR QUALITY

Figure 73 shows the result for the standard deviation of the velocity error in predicting the mean value for the several methods mentioned above as a function of the input signal-to-noise of the signal. One thousand spectral estimates were run to obtain each point. The ordinate of this figure should be multiplied by .795 to obtain the velocity error in meters per second. The results of this analysis are summarized as follows:

1. There is probably no significant difference in any of the methods at the extreme points ± 10 dB.

2. In between the extreme points, there is a region (near 5 dB) where the PPP method and the cascaded processor (ALPF followed by PPP) give the same result; there is a region (near 0 db) where the PPP is slightly better (about .5 db) than the cascaded processor; and there is a region (near -5 db) where the cascaded processor is somewhat better (about 1 db) than the PPP processor. The FFT (multi-lag order = 1) is the least desirable in all these cases, but serves as a standard of comparison.

The conclusion from the results is that optimum processing techniques may depend upon the signal-to-noise level. In such a case, a more detailed mapping of the various regions would be in order. This was not possible under the present contract because of the long run times required. However, these results indicate that the use of an adaptive filter as a part of the processing chain will lead to an improvement over the PPP processor of around one db for the pulsed LDV system near -5 db.

A remaining question which must be addressed concerns the convergence time of the adaptive filter. There is no theoretical work available at this time on this question for narrow band random processes, to the author's knowledge. These simulations have been guided by results for sinusoidal signals. It must yet be established that an adaptive filter can adapt using the amount of stationary data available in one or several LDV pulses.

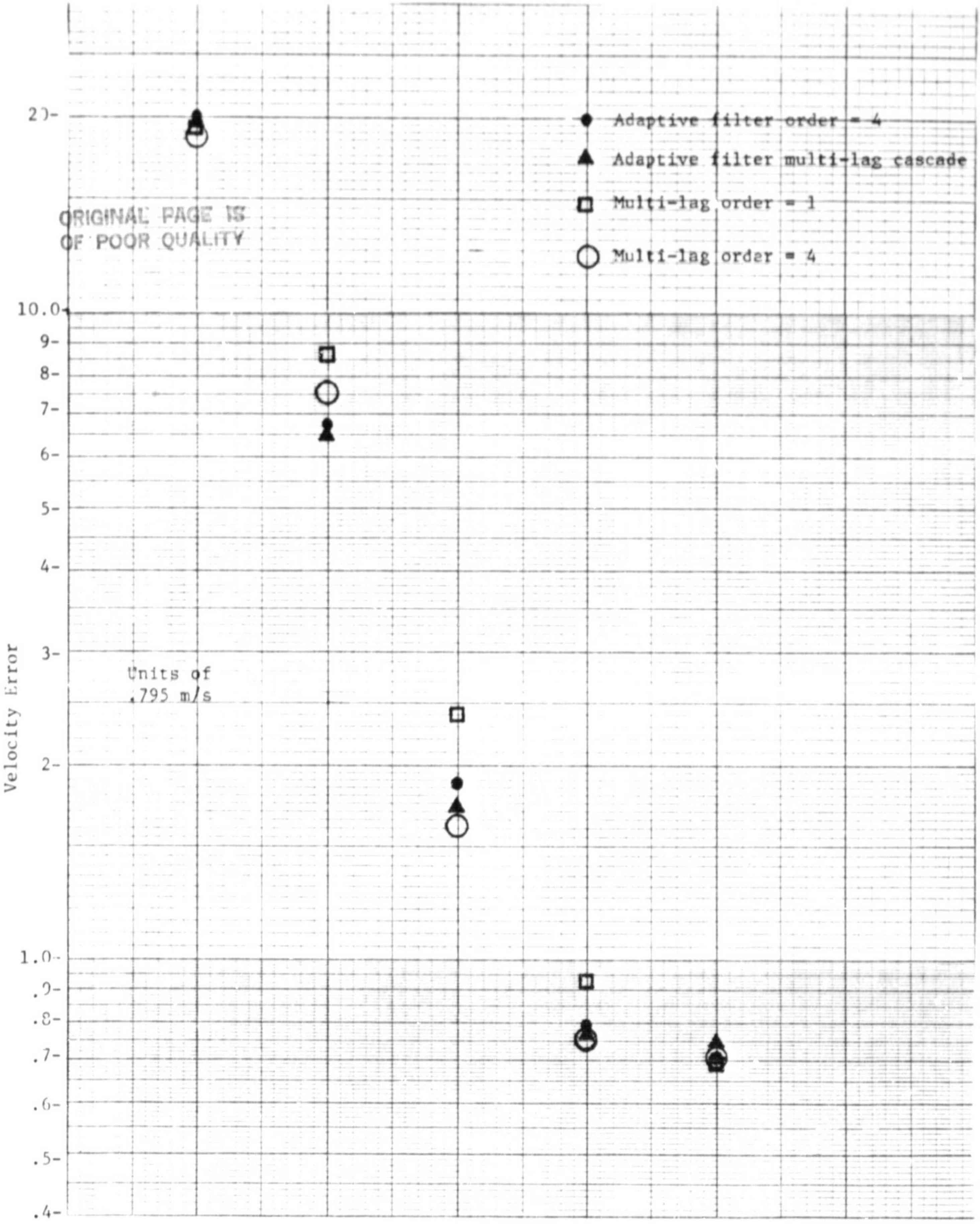


Fig. 73 Simulation Results

APPENDIX A. DATA PROCESSING CODE

```
C ****************************************************C
C
C  SET F:5 M;IN
C  SET F:6-ME;OUT
C  SET F:10-ME;OUT
C  SET F:12/CALFIL4;IN
C
C ****************************************************C
      COMMON/OPT/DS(22),NP(22),FSI,FN
      COMMON/COE2/SIG(22),SGL(22),UI
         INTEGER BUF, DATA
      DIMENSION DATA(256),ISBIN(256),BUF(384),CAL(26,51)
      DIMENSION IXBUF(26),XNOISE(51),YBUF(51),XSGL(22)
      DIMENSION ISN(256), XDS(22)
      DATA XSGL/-57.75,-56.0,-54.0,-50.25,-46.125,-41.5,-35.75/
      DATA XDS/4,8,16,32,64,127,1/,ITHRESH/4/,M/6/,MS/6/
C
C       IREC IS THE DATA RECORD COUNTER
       IREC=0
      DO 100 I =1,51
      CALL  BUFFERIN(12,1,IXBUF,26,ISTST,NURD,I=0)
C     WRITE(6,137) (IXBUF(IJ),IJ=1,26)
137   FORMAT(1H ,'CAL  DATA ',3(1X,9I8,/))
      DO 110 J=1,26
C
C     CAL -  THIS IS THE CALIBRATION DATA ARRAY
C             1ST INDEX IS OVER IF GAIN - 26 VALUES
C            2ND INDEX IS OVER DBSM VALUES RANGING FROM -71 TO -20
C
      CAL(J,I)  = IXBUF(J)
110   CONTINUE
100   CONTINUE
      DO 122 J=1,51
      XNOISE(J)=-71+J
122   CONTINUE
      KBUF = 0
      KH = 0
      KS = 0
      KF = 0
      KKK = 0
      MNSUM = 0
      SSVOL = 0
      NSUM = 0
      SVOL = 0
      ISUM =0
      DO 40 I=1,256
      ISN(I) = 0
40    ISBIN(I)=0
C
C     GETDAT READS THE DATA AND UNPACKS THE DATA
C     ARRAY TO CONFORM TO ORIGINAL DATA-TAKING
C

C       DROP OUT DATA IF AIRSPEED IS LESS THAN 150,
C       BECAUSE LDV USES THIS VALUE TO TUNE TO
C         THEREFORE DATA MAY BE BAD
C       DROP   OUT DATA IF THE COUNT IN THE
C         LAST BIN IS GREATER THAN 5000
C         THIS MEANS THE INVERSION ALGORITH
C         CANNOT HANDLE THESE CASES
C
5     CALL  GETDAT(BUF,DATA)
       IF(DATA(256).GT.5000) GO TO 5
```

```
      IF(DATA(1).EQ.1E6) STOP
        IREC = IREC + 1
        IF(BUF(32) .GT. 150) GO TO 175
6      CONTINUE
        CALL GETDAT(BUF,DATA)
      IF(DATA(1).EQ.1E6) STOP
      IREC=IREC+1
        GO TO 6
175     CONTINUE
      IF (KKK .GT. 0) GO TO 173
       KSH = BUF(5)
       KSM = BUF(6)
        KSS = BUF(7)
        KSF = BUF(8)
173      CONTINUE
        IF (BUF(8) .LE.BUF(4)) GO TO 69
        BUF(3) = BUF(3) - 1
        BUF(4) = BUF(4) + 1000
69      NF = BUF(4) - BUF(8)
        IF (BUF(7) .LE. BUF(3)) GO TO 71
        BUF(2) = BUF(2) - 1
        BUF(3) = BUF(3) + 60
71      NS = BUF(3) - BUF(7)
        IF (BUF(6) .LE. BUF(2) ) GO TO 73
        BUF(1) = BUF(1) - 1
        BUF(2) = BUF(2) + 60
73      NM = BUF(2) - BUF(6)
        NH = BUF(1) - BUF(5)
        KS = KS + (3600 * NH) + (60 * NM) + NS
        KF = KF + NF
        IF (KF .LT. 1000) GO TO 74
        KF = KF - 1000
        KS = KS + 1
74      CONTINUE
        KBUF = KBUF + BUF(32)
        KKK = KKK + 1
      DO 50 I=1,256
      ISUM=ISUM  + DATA(I)
50    ISBIN(I) = ISBIN(I) +DATA(I)
      ITR = ISL(BUF(15),16)
      UB = IOR(ITR,BUF(16))
      IFG = BUF(25)
      NSUM = NSUM + BUF(17)
      SVOL = SVOL + UB
       INOISE = 0
C
C     ISUM - THE TOTAL # OF PARTICLE HITS TO PROCESS
C     INOISE - THE NOISE SUM WHICH IS SUBTRACTED
C             FROM THE TOTAL # OF HITS
C
      CALL  GETDAT(BUF,DATA)
       IF(DATA(1).EQ.1E6) STOP
       IREC=IREC+1
      DO 60 L=1,256
      ISN(L) = ISN(L) + DATA(L)
60    INOISE = INOISE + DATA(L)
       ISUM=ISUM-INOISE
      ITR = ISL (BUF(15),16)
      UB = IOR (ITR,BUF(16))
      IFQ = BUF(25)
      NNSUM = NNSUM + BUF(17)
      SSVOL = SSVOL + UB
C      OUTPUT ISUM
```

```
C     OUTPUT ISUM,INOISE,ISBIN(256)
      IF (ISUM .LE. 1000)  GO TO 5
C
C      10000 PARTICLE COUNTS OR ABOVE MUST
C      BE OBTAINED BEFORE DATA IS SENT TO
C      INVERSION ALGORITHM
C
      KCH = BUF(1)
      KCM = BUF(2)
        KCS = BUF(3)
      KCF = BUF(4)
       IF(KSF .LE. KCF) GO TO 269
       KCS = KCS - 1
       KCF = KCF + 1000
269    IFRAC = (KCF-KSF)/2
       IF (KSS .LE. KCS) GO TO 270
       KCM = KCM - 1
       KCS = KCS + 60
 
270      NSECS = (KCS -KSS)/2
         IF (KSM .LE. KCM) GO TO 271
         KCH = KCH - 1
         KCM = KCM + 60
271      MMM = (KCM - KSM)/2
         MHR = KCH - KSH
        LF = KSF +IFRAC
       IF (LF. LT. 1000) GO TO 331
       LF = LF - 1000
       KSS = KSS + 1
331    LS = KSS + NSECS
       IF (LS .LT. 60) GO TO 332
       LS = LS - 60
       KSM = KSM + 1
332    LM = KSM + MMM
       IF (LM .LT. 60) GO TO 334
       LM = LM - 60
       KSH = KSH +1
334    LH = KSH + MHR
       OUTPUT IREC
       WRITE(6,138) VB,IFG,BUF(17)
138       FORMAT(1X,'VB,IFG,BUF(17)= ',F10.1,2I10)
      DO 200 I=1,51
      YBUF(I) = CAL(71-IFG,I)/500
200   CONTINUE
C
C     VB - THE ACCUMULATED VOLUME BETA COUNT  IN THE NOISE
C          DATA
C     VSIGL - INTEGRATED SIGNAL        OBTAINED
C             AFTER TABLE LOOKUP
C             AND AFTER NOISE HAS BEEN SUBTRACTED
C       RVB - VOLUME BETA(AVG) COUNT FOR THE DATA
C      VNS - INTEGRATED NOISE OBTAINED AFTER
C            TABLE LOOKUP
C     FINALB - THE FINAL VOLUME BETA FOR THIS DATA
C              SET
C
C     THE INTEGRATED VOLUME BETA(RVB) IS MULTIPLIED BY A
C     BIDIRECTIONAL REFLECTANCE PARAMETER(  = .016)
C     THEN DIVIDED BY THE SIGNAL RETURN FOR A
C     SANDPAPER DISK AT 10 METERS AT THE WAIST
C     ( = 1.8428924E6) AND THEN DIVIDED BY THE L-EFF
C     AT 10 METERS ( = 0.6407). SEE APPLIED
C     RESEARCH REPORTS FOR COMPLETE DOCUMENTATION.
```

```
C
C
      VB = SSVOL/NNSUM
      DBMNS = YLIN(51,VB,YBUF,XNOISE)
       RVB=SVOL/NSUM
       SDBMN=YLIN(51,RVB,YBUF,XNOISE)
       VSIGL=10**(SDBMN 10.)-10**(DBMNS/10.)
       UNS=10**(DBMNS/10.)
       FINALB=VSIGL/UNS*.016/1.042892 4E6/.6407
C      OUTPUT DBMNS
        OUTPUT FINALB,VSIGL,UNS,SSVOL,SVOL,NSUM
C
C     VB IS USED TO EXTRACT A NOISE VALUE FOR
C      THIS SET OF SINGLE PARTICLE DATA
C      BY INTERPOLATING
C     THE CALIBRATION DATA (CAL(I,J))
C
C     DBMNS - THE NOISE IN DBMS
C     ITHRESH - THE BIN WHERE THE SIGNALS ARE
C              THRESHOLDED - ABOVE ITHRESH CONTAINS
C              DATA
C    NP - THE NUMBER OF PARTICLES ARRAY -
C         CONTAINS NUMBER OF PARTICLE HITS IN THE
C         SELECTED BINS
C
C    THE FACTOR 1.5/.86 APPEARS IN THE NOISE
C    CALCULATION(DBMNN) BECAUSE THIS IS THE
C    DIFFERENCE IN BANDWIDTH BETWEEN THE SINGLE
C    PARTICLE DATA AND THE VOLUME CHANNEL DATA
C
       CNV=10**(DBMNS/10)
       DBMNN=10*LOG10(CNV*1.5/.86)
      DO 210 I = 1,M+1
      SGL(I) = 10**((XSGL(I)-DBMNN)/10)
  210 CONTINUE
C      OUTPUT SGL
      K =ITHRESH + 1
      DO 66 JJ=1,22
      NP(JJ) = 0
   66 CONTINUE
       DO 55 J=1,M+1
      KK=K+XDS(J)-1
      DO 10 I = K,KK
      NP(J) = NP(J) + ISBIN(I)-ISN(I)
   10    CONTINUE
      K=KK+1
   55 CONTINUE
C      OUTPUT NP,ISBIN,ISN
C
C
C     DETERMINE SIGMAS THAT CORRESPOND TO SIGNAL BINS
C     1.E8- CONVERTS FROM CM**2      TO MICRON**2
C     1.10376E7  IS THE SINGLE PARTICL GAIN AT AN AREA OF ZERO
      DO 300 I = 1,M  + 1
       SIG(I) = SGL(I)*1.E8/1.10376E7
  300 CONTINUE
      SIL = SIG(1)
      SIH = SIG(M  + 1)
C      OUTPUT SIG
        OUTPUT SIL,SIH
      SIH =SIG(M+1)
       OUTPUT ISUM
C
```


C-2

```
C      DATANAL DOES ACTUAL INVERSION
C
       CALL DATANAL(ISUM,M,MS,US,UV)
       ELS = KS + KF/1000.
       FLTL = KBUF/KKK*.5144*ELS
         IF(UV.EQ.0) UV=1.E 30
       SINGLEB = ISUM/4/3.14159265356/FLTL/UV*US
       OUTPUT SINGLEB,FLTL,ELS
C
C      ELS - THE ELAPSED TIME IN SECS FOR WHICH DATA
C            WAS TAKEN
C      FLTL - THE FLIGHT LENGTH IN METERS
C      SINGLEB - THE SINGLE PARTICLE BETA
C       US - THE AVERAGE CROSS-SECTION OF PARTICLE
C            SQUARE MICRONS
C        UV - THE AVERAGE SIZE OF THE TRANSVERSE
C             CROSS-SECTION OF FOCAL VOLUME
C             PREDICTED FROM INVERSION ALGORITHM
C
       WRITE(6,702) BUF(9),BUF(10), BUF(11)
 702    FORMAT(1X,'DATE ',I2,'-',I2'-',I2)
       WRITE (6,703) LH,LM,LS,LF
 703   FORMAT(1X,'TIME',I2,':',I2,':',I2,'.',I3)
       WRITE(6,704)
 704   FORMAT(/////)
         WRITE(13) LH,LM,LS,LF,SINGLEB,FINALB,US,FLTL,ELS,UV
       ISUM = 0
       INOISE=0
       GO TO 122
 878      STOP
       END
C
C
C
        SUBROUTINE GETDAT(BUF,DATA)
        DIMENSION DUM(7),DUMM(31),ID(20),DATA(256),BUFF(192),BUF(384)
        IMPLICIT INTEGER(C,S,D,F,V,B)
          III=0
  10        CONTINUE
          III=III+1
        II=0
 301    FORMAT(1X,6HBUFF= ,48(1X,8I8,/))
        CALL BUFFER IN(5,1,BUFF,192,ISTAT,NURD,IAB)
C       OUTPUT ISTAT,NURD,IAB
C        WRITE(6,301)  (BUFF(I),I=1,384)
         IF(ISTAT.EQ.3) DATA(1)=1E6
         IF(ISTAT.EQ.3) RETURN
        DO 300 I=1,384,2
        II=II+1
        I1=IAND(BUFF(II),8ZFF000000)
        I2=IAND(BUFF(II),8Z00FF0000)
        I3=IAND(BUFF(II),8Z0000FF00)
        I4=IAND(BUFF(II),8Z000000FF)
        I1=ISL(I1,-24)
        I2=ISL(I2,-8)
        I3=ISL(I3,-8)
        I4=ISL(I4,8)
        BUF(I)=IOR(I2,I1)
        BUF(I+1)=IOR(I4,I3)
 300      CONTINUE
        DO 100 K=1,20
        ID(K)=BUF(K+48)
 100    CONTINUE
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      DO 200 L=1,256
      DATA(L)=BUF(L+128)
200   CONTINUE
C     ************************************************************
C     CH = CURRENT TIME HOURS
C     CM = CURRENT TIME MINUTES
C     CS = CURRENT TIME SECONDS
C     CF = CURRENT TIME FRACTIONS
C     SH = START OF RECORD HOUR
C     SM = START OF RECORD MINUTE
C     SS = START OF RECORD SECOND
C     SF = START OF RECORD FRACTION
C     DBT = NUMBER BEFORE DIS. THRESHOLD
C     DT  = DISCRIMINATOR THRESHOLD
C     FBI = NUMBER BEFORE FORSARD/AFT IDENTIFIER
C     FI  = FORWARD/AFT IDENTIFIER
C     DATA = DATA
C     ID =  IDENTIFICTION STRING
C     ********************************************************
       CH=BUF(1)
       CM=BUF(2)
       CS=BUF(3)
       CF=BUF(4)
       SH=BUF(5)
       SM=BUF(6)
       SS=BUF(7)
       SF=BUF(8)
       DBT=BUF(23)
       DT=BUF(24)
       FBI=BUF(25)
       FI=BUF(26)
C     WRITE(6,401) SH,SM,SS,SF
401    FORMAT(1X,'START TIME: ',4I10)
C     WRITE(6,402) CH,CM,CS,CF
C    WRITE(6,405) BUF(15),BUF(16),BUF(17)
405  FORMAT (1X,'VOLUME BETA INT',3I10)
402     FORMAT(1X,'END TIME: ',4I10)
C     WRITE(6,403) DT
403    FORMAT(1X,'DISCRIMINATOR THRESHOLD ',I10)
C    WRITE(6,406) BUF(25)
406  FORMAT (1X,'IF GAIN',I10)
C     WRITE(6,404) FI
404    FORMAT(1X,'FORWARD/AFT ID ',I10)
C     WRITE(10) (ID(I),I=1,20)
C     WRITE(6,120) (DATA(I),I=1,128)
120   FORMAT(1X,16(1X,B18,/))
      IF (III.GT.500) RETURN
      RETURN
      END
C
C
     SUBROUTINE DATANAL(NT,M,MS,WS,WV)
      DIMENSION DAND(161,2)
      DIMENSION YY(125),ARR(126)
      COMMON/OPT/DS(22),NP(22),FSI,FN
      COMMON/ARE/F(2),AL,RD,R1,A
      COMMON/[illegible]/AR(101),DSR,SRMX,AXX(101)
      COMMON/[illegible] SIG(22),SGL(22),WI
      COMMON/[illegible]),Y(2),Z(2)
      DATA PW,ETA,BW,RO,F,AL,AH,NI,FSI/5.,.05,
    1 1.E+6,.046,10.,10.,1.,3.1622777,30,1.973/
     DATA PI,HP,CL,FLDA
    1/3.14159,6.62562E-34,3.E+8,10.6E-6/
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      DATA B,IP/1.5,1/
      DATA ARR/31*2.4E5,2.39E5,2.38E5,2.37E5,2.36E5,
     1 2.35E5,2.34E6,2.32E5,2.31E5,2.30E5,8.30E5,2*2.29E5,
     1 2.28E5,2.27E5,2.25E5,2.22E5,2.20E5,2.19E6,2.16E5,
     1 2.12E5,2.11E5,2.10E5,2.1E5,2.05E5,2.01E5,1.98E5,1.92E5,
     1 1.89E6,1.85E5,1.80E5,1.76E5,1.72E5,1.69E5,1.65E5,
     1 1.60E5,1.57E5,1.53E5,1.50E5,1.46E5,1.43E5,1.38E5,1.36E5,
     1 1.32E5,1.25E5,1.19E5,1.13E5,1.07E5,1.03E5,9.7E4,
     1 9.3E4,8.9E4,8.6E4,8.3E4,7.7E4,7.5E4,7.3E4,6.9E4,
     1 6.6E4,6.2E4,5.6E4,5.2E4,4.8E4,4.5E4,4.2E4,3.9E4,3.7E4,
     1 3.3E4,2.9E4,2.6E4,2.35E4,2.16E4,1.95E4,1.8E4,1.75E4,
     1 1.52E4,1.40E4,1.3E4,1.2E4,1.11E4,1.03E4,9.7E3,9.2E3,
     1 8.6E3,8.1E3,7.5E3,7.2E3,6.3E3,5.6E3,5.0E3,4.5E3,
     1 4.0E3,3.5E3/
      DATA YY/0,0.01,0.02,0.03,0.04,0.05,0.06,0.07,0.08,0.09,
     A  .10,.11,.12,.13,.14,.15,.16,.17,.18,.19,.20,
     A  .21,.22,.23,.24,.25,.26,.27,.28,.29,.30,
     A .32,.34,.36,.38,.40,.42,.44,.46,.48,.50,.52,.54,.56,
     A .58,.60,.65,.70,.75,.80,.85,.90,.95,1.0,1.1,1.2,1.3,
     A 1.4,1.5,1.6,1.7,1.8,1.9,2.0,2.1,2.2,2.3,2.4,2.5,
     A 2.6,2.7,2.8,2.9,3.0,3.2,3.4,3.6,3.8,4.0,4.2,4.4,4.6,
     A 4.8,5.0,5.2,5.4,5.6,5.8,6.0,6.5,7.0,7.5,8.0,8.5,9.0,
     A 9.5,10.0,11.,12.,13.,14.,15.,16.,17.,18.,19.,20.,21.,
     A 22.,23.,24.,25.,26.,27.,28.,29.,30.,32.,34.,36.,38.,
     A 40.,42./
      DATA TRANS/45.99/
      DATA N/3E6/
      DATA ITEST/1/
      IF (ITEST .EQ. 0) GO TO 744
      DO 999 I=1,123
      ARR(I) = (ARR(I) * TRANS)   * 1E-8
  999 CONTINUE
C
C    TRANS IS THE MULTIPLICATIVE FACTOR THAT CORRECTS
C     THE AREA CURVE
C      SO THAT THE AREAS CORRESPOND TO THE PROPER S/N/SIGMA
C
C    1.E-4 IS USED TO CONVERT THE AREAS FROM CM**2 TO M**2
C
      DO 998 I = 1,123
      YY(I) = YY(I) * 1E-4
  998 CONTINUE
      DO 77 I = 1,123
      J = 124 - I
      IF(I.GT.J) GO TO 78
      ATEMP = ARR(I)
      ARR(I) = ARR(J)
      ARR(J) = ATEMP
      YTEMP = YY(I)
      YY(I) = YY(J)
      YY(J) = YTEMP
   77 CONTINUE
   78   CONTINUE
      ITEST = 0
  744 A1 = AX*FLDA*FLDA*ETA/BW
      A=A1/(16.*PI*PI*PI)*1.E-12
      RD=PI*RO*RO/FLDA
      ALP=RD/F(1)
      R1=RO/2./RD
      REO1=F(1)*R1
      REO2=F(2)*R1
      G1=A/(REO1*REO1*REO1*REO1)
      G2=A/(REO2*REO2*REO2*REO2)
```

```
      DO 4 I = 1,M
      DS(I) = SIG(I+1) - SIG(I)
4     CONTINUE
      X(1)=0.
      Y(1)=0.
      Z(1)=0.
      SIL = SIG(1)
      SIH=SIG(M+1)
      SRMN=SGL(1)/SIH
      SRMX=SGL(1)/SIL
      ALS=1.+ALP*ALP
      G=SQRT(SRMX/SRMN)
      SQ=SQRT((G-1.)*ALS+1.)
      DSR=(SRMX-SRMN)/100.
      NPTS = 101
C           OUTPUT ARR,YY
      DO 2 I=1,NPTS
      J = NPTS - I + 1
      SR = SRMX - DSR * (I-1)
       AXX(J) = SR
       AR(J) = YLIN(123,SR,ARR,YY)
2     CONTINUE
C     OUTPUT AXX,AR
C     WRITE(103)(DAND(I,1),I=1,101),(DAND(I,2),I=1,101)
C     GOTO 65
C     CALL AREA(AT,SRMN)
C     CALL AREA1(AT1,SRMN)
C     OUTPUT AT,AT1
C     OUTPUT RD,ALP,ALS,G,SQ,F(1)
C     OUTPUT SRMN,SRMX,(SIG(I),I=1,MS+1),(SGL(I),I=1,M+1),XL,YL,ZL
       CALL OPTIM(B,NT,M,MS,NI,IP,WS,WV,YY,ARR)
      TSIA=NI*WS
      OUTPUT WV,WS
C     OUTPUT TSIA
      IF(WV.EQ.0.) WV=1.E-30
      BS =NT*WS/WV
      BM =4.*SMP*FLDA/A1
  479 FORMAT(6I12)
  489 FORMAT (12F8.3)
      RETURN
 65   END
C
C
      SUBROUTINE SPS(S,SI,J)
      COMMON/ARE/F(2),AL,RD,R1,A
      COMMON/SS/X(2),Y(2),Z(2)
      IF(J.EQ.1)X(1)=0
      RS=X(J)*X(J)+Y(J)*Y(J)
      ALP=RD/(F(J)+Z(J))
      GA=1./SQRT(1+(ALP*Z(J)/F(J))**2)
      RE=(F(J)+Z(J))*R1/GA
      E=RS/RE/RE
      FE=EXP(-E)
      SII=S/FE
      SI=(SII*RE**4)/A
      RETURN
      END
C
C
      SUBROUTINE AREA1(ARA,SR)
       COMMON/ARE1/AR(101),DSR,SRMX
      W=(SRMX-SR)/DSR
      NI=W
```

```
      IF(NI.GT.99)    GO TO 1
      Z=U-NI
      ARA=AR(NI+1)*(1-Z)+Z*AR(NI+2)
      GO TO 3
    1  CONTINUE
    2 FORMAT($AREA TABLE MAX$)
      ARA=AR(101)
    3 RETURN
      END
C
C
      SUBROUTINE AREA(ARA,S)
       COMMON/ARE/F(2),AL,RD,R1,A
       COMMON/ARE1/AE(101),DSR,SRMX
      ALS=1.+AL*AL
      G=SQRT(SRMX/S)
      G=G-1
      IF(G.LT.0.) OUTPUT G,S;GO TO 3
      SQ=SQRT(G*ALS+1.)
      ZM=(SQ+1.)*F(1)/ALS
      DEL=ZM/F(1)
      DZ=ZM/100.
      ARA=0.
      IT=0
      DO 1 I=1,204
      ZA=ZM-DZ*(I-2)
      ALP=RD/(F(1)+ZA)
      GA=1./SQRT(1.+(ALP*ZA/F(1))**2)
      RE=(F(1)+ZA)*R1/GA
      AR=A/(S*RE*RE*RE*RE)
      FL=LOG(AR)
      IF(FL.LT.0.) IT=1;GO TO 1
      ARA=RE*SQRT(FL)+ARA
    1 CONTINUE
      ARA=ARA*2.*DZ
      IF(IT.EQ.0) WRITE(108,2)
    2 FORMAT($INTEGRATION RESTRICTED$)
    3 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE SSP(S,SI,J)
       COMMON/ARE/F(2),AL,RD,R1,A
       COMMON/SS/X(2),Y(2),Z(2)
      IF(J.EQ.1) X(1)=0.
      RS=X(J)*X(J)+Y(J)*Y(J)
      ALP=RD/(F(J)+Z(J))
      GA=1./SQRT(1+(ALP*Z(J)/F(J))**2)
      RE=(F(J)+Z(J))*R1/GA
      S=A*SI/(RE*RE*RE*RE)
      E=RS/RE/RE
      FE=EXP(-E)
      S=S*FE
      RETURN
      END
C
C
      SUBROUTINE COEF2(C,I,J,K,DJ,M)
      COMMON/COE2/SIG(22),SGL(22),UI
      COMMON/ARE1/AR(101),DSR,SRMX,AXX(101)
      A2=0.
      SI=SIG(J)+(K-1)*DJ
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SR=SQL(I)/SI
      A1 = YLIN(101,SR,AXX,AR)
      IF(I.GT.M+1) GO TO 1
      IF(I.EQ.J) GO TO 1
      SR=SGL(I+1)/SI
      A2 = YLIN(101,SR,AXX,AR)
    1 C =A1-A2
      RETURN
      END
C
C
      SUBROUTINE LPICK(SIG,MS,BP1,BP,DB,ALP,DALP)
      DIMENSION SIG(22)
      DB=(SIG(MS+1)-SIG(1))/20.
       BP=(SIG(1)+SIG(MS+1))/2.
      A1=SQRT(LOG(SIG(MS+1))/2./SIG(1)*LOG(SIG(1))*.3/2./SIG(1)
     1 /LOG(SIG(MS+1))-LOG(SIG(1))/SIG(1))
      DALP=A1/20.
       ALP=A1/2.
       RETURN
       END
C
C
      SUBROUTINE EPICK(SIG,BP1,BP,MS,DP)
      DIMENSION SIG(22)
      S1=BP1*LOG(SIG(1))/SIG(1)
      S2=BP1*LOG(SIG(MS+1))/SIG(MS+1)
      BP=S2
      DP= (S1-S2)/25.
      RETURN
      END
C
C
       SUBROUTINE PROW(BP,IP,SIP,M)
        COMMON/COE2/SIG(22),SGL(22),U
      COMMON/ALN/S0,A3
C     OUTPUT   IP,BP,SIP
       IF(SIP.LE.0.) OUTPUT SIP
       IF(SIP.LE.0) U=0
       IF(SIP.LE.0) RETURN
      GOTO (1,2,3)IP
    1 X=1.-BP
        A1=-(SIG(1)**X-SIG(M+1)**X)/X
        IF(BP.GT..999.AND.BP.LT.1.001)A1=LOG(SIG(M+1)/SIG(1))
        U=(SIP**(-BP))/A1
        RETURN
    2   A2=(EXP(-BP*SIG(1))-EXP(-BP*SIG(M+1)))/BP
        U=EXP(-BP*SIP)/A2
        RETURN
    3  U=(EXP(-(LOG(SIP)-LOG(BP))**2/2./S0**2))/A3
       RETURN
        END
C
C
      SUBROUTINE OPTIM(B,N,M,MS,NI, IP,USS,UUS,YY,ARR)
      DIMENSION PN(22)
      DIMENSION ARR(125),YY(125)
      COMMON/OPT/DS(22),NP(22),FSI,FN
      COMMON/COE2/SIG(22),SGL(22),UI
       COMMON/ALN/S0,A1
      COMMON/ARE1/AR(101),DSR,SRMX,AXX(101)
      OUTPUT IP,B
      B=2.5
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      NC=0
       IP=1
      IC=0
      BP=B
       IT=0
        BP=B
        DB=.1
   10 CONTINUE
      II=0
      RM=1.E+60
       IF(NC.GT.1)IC=1
      IF(IC.EQ.1)NC=0
       IP=2
      IF(IC.EQ.0)IP=1
       IF(IT.GE.2)IP=3
        IC=0
C
C        TEMPORARILY THIS DEFEATS THE EXP AND LOG-NORMAL SOLNS
C
        IF(IP.EQ.2) RETURN
       IF(IP.EQ.2) CALL EPICK(SIG,BP1,BP,MS,DB)
       IF(IT.EQ.2)CALL LPICK(SIG,MS,BP1,AM,DB,S0,DALP)
    8   CONTINUE
      II=II+1
      IF(II.GT.100) RETURN
       IF(BP.LT.0..AND.IP.EQ.1) RETURN
        IF(IP.LE.2)GOTO 102
        AM=AM+DB
        IF(AM.LT.1.) DB=0.
       IF(AM.LT.1.) AM=ABS(AM/10)
        IF(IT.EQ.3)S0=S0+DALP
        IF(IT.EQ.3)AM=BP1
C
C        NORMALIZATION OF LOG-NORMAL DISTRIBUTION
C
        A1=0.
        DO 101 J=1,MS
        DJ=DS(J)/NI
        DO 100 K=1,NI+1
       SIP=SIG(J)+(K-1)*DJ
        W=EXP(-(LOG(SIP)-LOG(AM))**2/2./S0**2)
  100   A1=A1+W*DJ
        SI1=SIG(J)
        SI2=SIG(J+1)
        U1=EXP(-(LOG(SI1)-LOG(AM))**2/2./S0**2)
        U2=EXP(-(LOG(SI2)-LOG(AM))**2/2./S0**2)
  101   A1=A1-.5*(U1+U2)*DJ
        BP=AM-DB
C      OUTPUT S0,A1
  102   CONTINUE
    6 DO 5 I=1,MS
    5 PN(I)=0.
      BP=BP+DB
        IF(IT.EQ.3)BP=BP1
      OUTPUT BP
      US=0.
      WV=0.
       DO500 J=1,MS
       DJ=DS(J)/NI
      DO 501 K=1,NI +1
       SIP=SIG(J)+(K-1)*DJ
      SR=SGL(1)/SIP
      ASK=YLIN(101,SR,AXX,AR)
```

```
      CALL PROW(BP,IP,SIP,MS)
       US=US+WI*SIP*DJ
 501   UU=UU+WI*ASK*DJ
       SI1=SIG(J)
       SI2=SIG(J+1)
       SR=SGL(1)/SIG(J)
      AS1=YLIN(101,SR,AXX,AR)
       SR=SGL(1)/SI2
      AS2=YLIN(101,SR,AXX,AR)
       CALL PROW(BP,IP,SI1,MS)
       U1=WI
       CALL PROW(BP,IP,SI2,MS)
       U2=WI
       US=US-.5*(U1*SI1+U2*SI2)*DJ
 500    UU=UU-.5*(U1*AS1+U2*AS2)*DJ
      US=US*1.E-12
      US5=US1
      UU5=UU1
      US1=US
      UU1=UU
      FN=N
      DNT=FN/UU
      BT=DNT*US
      DO 1 I=1,M+1
      DO 1 J=1,MS
      DJ=DS(J)/NI
      DO 4 K=1,NI+1
       SIP=SIG(J)+(K-1)*DJ
       CALL PROW(BP,IP,SIP,MS)
       CALL COEF2(C,I,J,K,DJ,M)
   4   PN(I)=PN(I)+C*WI*DJ
       SIP=SIG(J)
       CALL PROW(BP,IP,SIP,MS)
       U1=WI
       SIP=SIG(J)+NI*DJ
       CALL PROW(BP,IP,SIP,MS)
       U2=WI
       CALL COEF2(C1,I,J,1,DJ,M)
       CALL COEF2(C2,I,J,NI +1,DJ,M)
   1   PN(I)=PN(I)-.5*(C1*U1+C2*U2)*DJ
      R=0.
      DO 2 I=1,MS
      FN=NP(I)
      FP=PN(I)*DNT
C     OUTPUT R,FN,FP,I
       IF(I.LE.M) R=(FN-FP)**2+R
   2   CONTINUE
      NC=NC+1
C      OUTPUT NC,DNT
      IF(NC.NE.2)GOTO 11
      IF(R.LT.RM)GOTO 11
C       OUTPUT R,RM
      RM=R
      DB=-DB
       IF(IT.EQ.3) DALP=-DALP
      GOTO 8
  11  CONTINUE
      IF(R.LT.RM) RM=R;GO TO 8
  19  FORMAT( 5F10.0)
   29 FORMAT(F10.3,4E12.3)
   39 FORMAT(5F12.4)
   49 FORMAT(5F10.0)
      IT=IT+1
```

```
      IF(IT.EQ.1)RP=RM;USP=USS;UUP=UUS
      IF(IT.EQ.2)RE=RM;USE=USS;UUE=UUS
      IF(IT.GT.2)RLM=RM;USLM=USS;UULM=UUS
      BP1=BP-DB;OUTPUT BP1
      IF(BP.LT.0.) BP=0.
    7 BP=BP-DB
      IF(IT.LT.4) GOTO 10
      OUTPUT RP,RE,RLM
       OUTPUT USP,USE, USLM,UUP,UUE,UULM
      USS=USP
      UUS=UUP
      IF(RP.LT.RE.AND.RP.LT.RLM)RETURN
      USS=USE
      UUS=UUE
      IF(RE.LT.RLM)RETURN
       USS=USLM
       UUS=UULM
      RETURN
      END
C
C
      FUNCTION YLIN(M,XX,X,Y)
C     M - NUMBER OF DATA POINTS
C     XX - X VALUE FOR WHICH Y VALUE MUST BE INTERPOLATED
C     X - DEPENDENT VARIABLE ARRAY
C     Y - INDEPENDENT VARIABLE ARRAY FOR WHICH
C         INTERPOLATIONS ARE PERFORMED
      DIMENSION X(1),Y(1)
      J=1
      DO 10 I=J,M
      IF (XX .LT. X(I)) GO TO 11
   10 CONTINUE
      I=M
   11 IF (I .EQ. 1) I=2
      J=I-1
      P=(XX-X(J))/(X(I)-X(J))
      IF (P .LT. 0. .OR. P .GT. 1.) P = 0.0
      YLIN=Y(J) +P*(Y(I)-Y(J))
      RETURN
      END
*
```

APPENDIX B

```
C
C
C                     PROGRAM CESTIM
C
C
C                                  BY R. JEFFREY KEELER
C                                       RON A. RICHTER
C                                       KARL B. YOUNG
C                                       SHANE C. MORRISON
C                                       DAVE KCHURCHILL
C
C
C
C
C
C        SWITCH FUNCTIONS
C
C        0          PLOT TIME SERIES
C        1          PLOT FFT
C        2          PLOT MAX ENTROPY-BURG
C        3          PLOT MAX ENTROPY-YULE/WALKER
C        4          PLOT ADAP-MAX ENTROPY
C        5          PLOT ADAP-WEIGHT TRANSFORM
C        6          PLOT ADAP-PREDICTED DATA TRANSFORM
C        7          PRINT COMPLEX COVARIANCE SPECTRUM MOMENTS
C        8          PLOT TRANSFORM OF MULTI-LAG COVARIANCE FUNCTION
C        9          PLOT AVERAGED FFT
C        10-12      SPARE
C        13         PLOT NEW FREQUENCY AXES
C        14         DO HISTOGRAMS
C        15         STOP PLOTTING
C
C*****************************************************************
C
    1    COMMON/PKS/PEAKS(3,1024,8)
         COMMON/SPEC/ ISPCFL(7),ISPC
                 COMMON/SMR/RMS,SUMXI
           COMMON/LA/FLAGG
         COMMON NEST,NAV,IEST,IWNDW,LORD,ALPHA,WGTS(128),NQ
         COMMON/SW/ ISW(0:15)
         COMMON/SIGNL/ SIG(1024),RSLT(1024),WORK(1024),
        1NFTSF,NPTS,NSTRT,NTOT,NDAT,PSIG,SNRF,SNR,
        2ISYNDAT,GSIG,GNOISE,FREQ,SIGMA
         COMMON/PLS/ XNORM(10),ICT,LOGSCL,FIRST,IPXFL
          COMMON/LBLS/LABEL(10,8)
         COMPLEX SIG,WGTS,R(129),SIGDAN(1024),SAVE(12)
         EQUIVALENCE (R(1),RSLT(1))
         LOGICAL FIRST,ISW
         EXTERNAL OVFINIT,OVINIT,OVPROC
         ISW(0)=.TRUE.
```

```
      ISW(1)=.FALSE.
      ISW(2)=.FALSE.
      ISW(3)=.FALSE.
      ISW(4)=.FALSE.
      ISW(5)=.FALSE.
      ISW(6)=.FALSE.
      ISW(7)=.FALSE.
      ISW(8)=.FALSE.
      ISW(9)=.FALSE.
      ISW(10)=.FALSE.
      ISW(11)=.FALSE.
      ISW(12)=.FALSE.
      ISW(13)=.FALSE.
      ISW(14)=.FALSE.
      ISW(15)=.FALSE.
C
C                INITIALIZE PLOTTING PACKAGE)
C
        CALL INITT (900)
C
C                GET INPUT PARAMETERS
C
      CALL INPUT
C
C
C                INITIALIZE ARRAYS, OPEN FILES, ETC.
C
        CALL INIT
C
C                IF WE ARE EXAMINING AN OLD SPCMCM GO TO HISTOGRA
C
      IF (ISPC.EQ.1) GO TO 62
C
C                GO THROUGH THE DIFFERENT ESTIMATORS NEST (NUMBER
C                  ESTIMATIONS) TIMES.
C
          NDAT2=NDAT
          NPTSF2=NPTSF
      IS10=1
            FLAGG=0
            FIRST=.TRUE.
         DO 129 IDAI=1,LORD
  129       SAVE(IDAI)=WGTS(IDAI)
      DO 10 IRLZ=IS10,NEST
          NSTRT=IRLZ
C
C
C                GET DATA, AND STORE IT IN SIG
C
C                              NSEED=0--INDICATES THIS IS FIRST TIME TH
C                                NOT A REFRESH
C
```

```
            ININ=IRLZ
          NPTSF=NPTSF2
      CALL GETSIG (SIGPWR)
C
C
C                PLOT COMPLEX TIME SERIES
C
      DO 11 IDAI=1,2*NPTSF2
        SIGDAN(IDAI)=SIG(IDAI)
   11   CONTINUE
    6   TXO=5.
      IF (ISW(0)) CALL TIMGRF (SIGPWR,TXO)
C
C          NOW LET'S GO THROUGH THE DIFFERENT ESTIMATORS
C
      DO 20 IST=1,8
C

C                IF BIT IST IS NOT SET, ESTIMATOR IST WAS NOT REQUESTE
C
      IF (.NOT.ISW(IST)) GO TO 20
C
C                PERFORM THE ESTIMATION
C
      GO TO (100,200,300,400,500,600,700,800) IST
C
C*************************************************************************
C
C                FAST FOURIER TRANSFORM
C
C                THE FFT HAS SPECIAL PARAMETERS CONNECTED WITH IT WHIC
C                ALLOWS IT TO DO AN FFT ON ALL INPUT POINTS (NTOT)
C                EVEN IF THE FFT-SIZE (NPTS) FOR THE OTHER ESTIMATORS
C                REQUESTED TO BE SMALLER
C
C
  100   NDAT=64
       NPTSF=NPTSF2
      CALL SETFFT
C
C                GET POWER, FREQ,AND WIDTH OF SPECTRUM
C
      CALL SPKPKR (IST,IRLZ)
C
C
C                CHECK SWITCHES TO SEE IF YOU WANT A SPECTRUM PLOT
C
       IF (FIRST) CALL PLSPEC (IST)
C
C
C     UPDATE AVERAGE SPECTRUM
C
```

```
      CALL SPCAV ( IST,IRLZ)
C
C     PLOT IF DESIRED
C
      IF(ISW(9)) CALL PLSPEC ( IST)
C
C               END FFT--INPUT
C
      GO TO 40
C*****************************************************************************
C
C
C               MAXIMUM ENTROPY--BURG ALGORITHM
C
C
  200 CALL CBURG
C
C               THE REST OF THE BURG CALCULATION CAN BE DONE THE SAME
C                         AS THE OTHER MAXIMUM ENTROPY ESTIMATORS
C
      GO TO 30
C*****************************************************************************
C
C
C               MAXIMUM ENTROPY--YULE/WALKER ALGORITHM
C
C
C               GET CORRELATION COEFFICIENTS FIRST
C                 (STORE IN RSLT)
C
  300 CALL CCORR
C
C
C               THEN PROCESS THOSE COEFFICIENTS
C
      CALL CLEV
C
C               THE REST OF THE YULE/WALKER CALCULATION CAN BE DONE
C                         SAME AS THE OTHER MAX. ENTROPY ESTIMATORS
C
      GO TO 30
C*****************************************************************************
C
C
C               MAXIMUM ENTROPY--ADAPTIVE FILTER ALGORITHM
C
C
  400 CALL CALPF
C
C               SAVE THE PREDICTED OUTPUT IN 'XOUT'
C
C
```

```
C                  FROM HERE ON, 'TIS THE SAME FOR ALL MAX. ENTROPY
C                           ESTIMATORS
C
30    CONTINUE
C
      CALL MEMFFT
C
C                  JOIN THE REST OF THE ESTIMATORS FOR PEAK-PICKING,
C                  PLOTTING,ETC.
C
      GO TO 50
C*******************************************************************
C
C
C
C                           ADAPTIVE FILTER--WEIGHTS
C
C
C                  IF ALPF HAS ALREADY BEEN CALLED, THERE'S NO NEED TO DO
C                  IT AGAIN...
C
500   IF(ISW(4)) GO TO 510
      CALL CALPF
C
C                  SAVE THE PREDICTED OUTPUT IN 'XOUT'
C
C
C                  DO FOURIER ANALYSIS ON WEIGHTS
C
          DO 510 IDAA=1,LORD
510         SIG(IDAA)=WGTS(IDAA)
            NDAT=LORD
             NPTSF=NPTS
          DO 511 IDA=LORD+1,4*NPTSF
511          SIG(IDA)=0.
            CALL SETFFT
         FLAG=1.
      GO TO 50
C*******************************************************************
C
C
C                           ADAPTIVE FILTER--PREDICTED
C
C                  IF ALPF HAS ALREADY BEEN CALLED, THERE'S NO NEED TO DO
C                  IT AGAIN....
C
600          NDAT=NDAT2
             NPTSF=NPTS
        CALL CALPF
         IF (ISW(12))  GOTO 800
C
C                  DO FOURIER ANALYSIS ON PREDICTED OUTPUTS
```

```
C
620   CALL SETFFT
C
C
C**********************************************************************
C
C
C                    GET THE FIRST 3 MOMENTS OF SPECTRUM AND STORE THEM
C
50    CALL SPKPKP (IST,IRLZ)
C
C
C                    CHECK SWITCHES TO SEE IF YOU WANT A PLOT OF THE
C                    SPECTRUM
C
        IF (FIRST) CALL PLSPEC ( IST)
          IF(FLAG.EQ.1)CALL PLSPEC(IST)
          FLAG=0.
C
40    CONTINUE
C
C                    RESTORE SIGNAL
C
      DO 41 IDII=1,2*NPTSF2
       SIG(IDII)=SIGDAN(IDII)
41      CONTINUE
         DO 139 IDAI=1,LORD
139    WGTS(IDAI)=SAVE(IDAI)
      GO TO 20
C
C**********************************************************************
C
C                    ROUTINES FOR COMPLEX COVARIANCE
C
C
C                    CALCULATE COMPLEX COVARIANCE
C
700     LORD11=1
       CALL CCORR
C
C       CALL COMPLEX COVARIANCE ESTIMATOR
C
      CALL CCOV (IST,IRLZ)
C
C                    PLOT POW,FREQ,AND VAR HERE
C
C       IF (ISW(7)) OUTPUT IFP,ICT,' CC',RSLT(1),RSLT(2),RSLT(3)
       GO TO 20
C
C**********************************************************************
C
C                    ROUTINE FOR MULTI-LAG COMPLEX COVARIANCE
C
```

```
C                  CALCULATE COMPLEX COVARIANCE FOR LORD LAGS
C
  800     NDAT=NDAT2
          CALL CCORR
C
C                  GENERATE SYMMETRIC COVARIANCE FCT
C
          LORDP=LORD+1
          DO 805 I=1,LORDP
          SIG(I)=R(I)
  805          CONTINUE
          N1 = LORD+2
          N2 = NPTS - LORD
          DO 810 I=N1,N2
  810     SIG(I)=CMPLX(0.0,0.0)
          DO 820 I=1,LORD
          J=NPTS+1-I
          SIG(J)=CONJG(R(I+1))
  820          CONTINUE
C                  DO FFT ON COVARIANCE FUNCTION
C
             NDAT=NPTS
             NPTSF=NPTS
          CALL SETFFT
          GO TO 50
C
C                  END-OF-LOOP    20
C
  20      CONTINUE
C**********************************************************************
C**********************************************************************
C
C
C                  END-OF-LOOP    10
C
           ISW(1)=.FALSE.
           FIRST=.FALSE.
  10      CONTINUE
  60      CALL BELL
C
C                  HISTOGRAM PLOTTING SEQUENCE
C
C                  1-READ IN RAW DATA          (DSKRD)
C                  2-DO STATISTICS ON THEM     (STATS)
C                  3-GET HISTOGRAM FROM THEM   (FHIST)
C                  4-PLOT DATA AND HISTOGRAM   (HPLOT)
C
C
C                  CHECK WHICH VARIABLE TO DO HISTOGRAMS OVER
C
C              CALL HDCOPY
            CALL BELL
```

```
         INPUT DANNY
62    CALL ERASE
         CALL HOME
      CALL ANMODE
      OUTPUT 'DO HISTOGRAMS  FOR:'
      OUTPUT'0-STOP'
      OUTPUT '-1-RESTART'
      OUTPUT ' 1-POWER'
      OUTPUT ' 2-FREQUENCY'
      OUTPUT ' 3-WIDTH'
       INPUT IHIST
                 ISW(C)=.TRUE.
         CALL ERASE
       IF(IHIST.EQ.0)GOTO 9999
      IF ((IHIST.EQ.0).AND.(ISPC.EQ.1)) GO TO 62
      IF (IHIST.GT.3) GO TO 62
      IF (IHIST.LT.0) GO TO 1
C
C                 CYCLE THROUGH ALL HISTOGRAMS
C
      DO 70 IST=1,8
C
C
C                 IF WE DIDN'T USE THIS ESTIMATION
C                 GOING TO DO A HISTOGRAM ON IT.
C
      IF(.NOT.(ISW(IST))) GO TO 70
C
C                 1-READ IN DATA
C
      CALL DSKRD (IST,IHIST)
C
C                 3-GET HISTOGRAM FROM THEM
C
C                   (NOTE:  THE FFT MAY HAVE A DIFFERENT NUMBER
C                   OF POINTS TO DO A HISTOGRAM OVER)
C
      NBINS=NPTS
      IF (IST.EQ.1) NBINS=NPTSF
      IF(IHIST.NE.2) NBINS=100
      CALL EHIST (HMAX,NBINS,IHIST)
C
C                 2-DC STATISTICS ON THEM
C
      CALL STATS ( XMEAN,STDEV,NBINS)
C
C                 4-PLOT DATA HISTOGRAM
C
      DO 75 I=1,NEST
      SIG(I)=CMPLX(WORK(I),0.0)
75               CONTINUE
C
```

```
      CALL HPLOT (HMAX,XMEAN,STDEV,NBINS,IHIST,IST)
C
70    CONTINUE
      GO TO 62
9999   STOP
      END
C
C
C
C
C
C
      SUBROUTINE INPUT
C
C
C*************************************************************************
C
C             ROUTINE TO INPUT FROM KEYBOARD EVERYTHING IT
C               DOESN'T KNOW
C
C
C*************************************************************************
C
      COMMON/SPEC/ISPCFL(7),ISPC
      COMMON NEST,NAV,IEST,IWNDW,LORD,ALPHA,WCTS(128),NO
      COMMON /SW/ ISW(0:15)
      COMMON/SIGNL/ SIG(1024),RSLT(1024),WORK(1024),
     1NPTSF,NPTS,NSTRT,NTOT,NDAT,PSIG,SNRF,SNR,
     2ISYNDAT,GSIG,GNOISE,FREQ,SIGMA
      COMMON/PLS/ XNORM(10),ICT,LOGSCL,FIRST,IPXFL,LNAME(4),LPL
      COMMON/LBLS/LABEL(20,8)
      COMPLEX SIG,WCTS
      LOGICAL FIRST,ISW
      DIMENSION MTFILE(4),IHEAD(256),IDT(3),ITM(3)
      DATA ((LABEL(I,J),I=1,10),J=1,8)/'FAST',' FOU','RIER',
     1' TRA','NSFO','RM  ',4*'    ',
     1'MAXI','MUM ','ENTR','OPY-','-BUR','G AL','GORI','THM ',
     12*'    ',
     1'MAXI','MUM ','ENTR','OPY-', '-YUL','E/WA','LKER',
     1' ALG','ORIT','HM  ',
     1'MAXI','MUM ','ENTR','OPY-','-ADA','PTIV','E-FI','LTER',
     1' ALG','RTHM',
     1'ADAP','TIVE',' FIL','TER-','-WEI','GHTS',4*'    ',
     1'ADAP','TIVE',' FIL','TER-','-PRE','DICT','ED  ',3*'    ',
     1'COMP','LEX ','COVA','RIAN','CE  ',5*'    ',
     1'MULT','I-LA','G CO','MPLE','X CO','VARI','ANCE',3*'    '/
      EQUIVALENCE (IHEAD(1),WORK(1))
      IEST=0
C
C             GET SOME INFORMATION ON THE INPUT DATA
C
       CALL ANMODE
```

```
        ISPC=0
C
C                     NBRP-NARROW BAND RAYLEIGH PROCESS
C     DO WE WANT SINUSOIDAL OR NBRP OUTPUT OF SYNTHETIC
C
20    OUTPUT 'SINUSOIDAL(1) OR NBRP(2) SYNTHETIC DATA;'
      INPUT ISYNDAT
      OUTPUT ISYNDAT
C
C              GIVE THEM A CHOICE OF ESTIMATORS
C
40    CALL MOVABS (0,650)
      OUTPUT 'THE FOLLOWING ESTIMATORS ARE AVAILABLE SINGLY OR'
       OUTPUT 'SIMULTANEOUSLY.'
      OUTPUT 'INPUT A  1 IF YOU WANT THE INDICATED ESTIMATOR, 0 IF NOT'
      OUTPUT ' '
      DO 100 I=1,6
       WRITE(10,1000) (LABEL(J,I),J=1,10)
1000  FORMAT(20A4)
100   CONTINUE
C
C               NOW REPRINT NAME OF ESTIMATOR AND ACCEPT A 1 OR 0 AFTER
C               IT
C
      DO 200 I=1,6
      WRITE (10,1020) (LABEL(J,I),J=1,10)
1020  FORMAT (20A4,Z)
199      INPUT IST
      OUTPUT ' '
      IF (IST .EQ. 1)  ISW(I) = .TRUE.
200   CONTINUE
C
C                    EST 6 -- ADAPTIVE (FOURIER TRANSFORM) PREDICTED
C               IF ESTIMATOR 6 WAS SPECIFIED, WE NEED TO KNOW HOW
C               MANY OUTPUT DATA POINTS WE WANT TO
C               GET FROM THE ADAPTIVE FILTER
C
      IF (.NOT.ISW(6)) GO TO 300
        OUTPUT'DO YOU WANT TO ENTER MULTI-LAG FROM ALPF1-YES,0-NO'
        INPUT MDAN
        IF (MDAN.EQ.1)ISW(12)=.TRUE.
       OUTPUT '*NUMBER OF ALPF PREDICTION SAMPLES TO USE'
      INPUT NDAT
      OUTPUT NDAT
C
C               IF MAX ENTROPY ESTIMATORS WERE SPECIFIED, WE ASK FOR
C               THEIR ORDER...
C
300   OUTPUT ' '
      LORD=-1
      IF (.NOT.(ISW(2).OR.ISW(3))) GO TO 310
      OUTPUT 'ORDER OF MAXIMUM ENTROPY ESTIMATORS'
```

```
      INPUT LORD
      OUTPUT LORD
      GO TO 320
C
C                   IF MAX ENTROPY NOT SPECIFIED, BUT ADAPTIVE IS, THEN
C                     WE ASK FOR THEIR ORDER (IT'S THE SAME THING)
C
  310 IF (.NOT.(ISW(4).OR.ISW(5).OR.ISW(6))) GO TO 340
      OUTPUT 'ORDER OF ADAPTIVE FILTER'
      INPUT LORD
      OUTPUT LORD
      GO TO 330
C
C                   IF ADAPTIVE IS SPECIFIED, WE NEED SOME OTHER NEAT
C                   PARAMETERS
C
  320 IF(.NOT.(ISW(4).OR.ISW(5).OR.ISW(6))) GO TO 340
  330 OUTPUT 'NORMALIZED ADAPTIVE CONSTANT (ALPHA)  '
      INPUT ALPHA
      OUTPUT ALPHA
C
C                   IF MAX ENTROPY OR ADAPTIVE ESTIMATORS NOT
C                   SPECIFIED AND IF MULTI-LAG COMPLEX CAVARIANCE
C                   IS SPECIFIED, THEN GET NUMBER OF LAGS (ORDER)
C
  340 IF (LORD .EQ. -1 .AND. ISW(6)) GO TO 341
      GO TO 342
  341 OUTPUT 'ORDER OF MULTI-LAG COMPLEX COVARIANCE ESTIMATOR'
      INPUT LORD
      OUTPUT LORD
C
C                   IF SYNTHETIC DATA, WE NEED TO KNOW WHAT FREQUENCY,ETC
C
  342 IF (ISYNDAT.EQ.0) GO TO 350
      OUTPUT 'FREQUENCY OF SYNTHETIC DATA'
      INPUT FREQ
      OUTPUT FREQ
C
C                   AND IF IT'S NBRP DATA, WE NEED A WIDTH...
C
      IF (ISYNDAT .EQ. 2) GO TO 2323
   343 OUTPUT 'SNR OF SYNTHETIC DATA (DB)'
      INPUT SNR
      OUTPUT SNR
      GO TO 360
 2323 OUTPUT 'STANDARD DEVIATION OF SIGNAL'
      INPUT SIGMA
      OUTPUT SIGMA
      GO TO 343
C
C                   BUT IF ACTUAL DATA, WE NEED SOME OTHER PARAMETERS...
C
```

```
350   OUTPUT 'ESTIMATED WIDE-BAND SNR OF DATA (DB)',SNR
C
C              GET INFORMATION ABOUT FFT PROCESSING
C
360   NSTFT=1
362   IF (NSTFT .LT. 1) GO TO 360
      OUTPUT 'TOTAL NUMBER OF INPUT DATA POINTS'
      INPUT NTOT
      OUTPUT NTOT
      ISPCPLT=0
      IF (ISW(1).OR.ISW(2).OR.ISW(3).OR.ISW(4).OR.ISW(5).OR.ISW(6)
     +.OR.ISW(8)) ISPCPLT=1
      IF (ISPCPLT .EQ. 1) GO TO 367
      GO TO 369
367   OUTPUT 'LENGTH OF TRANSFORM FOR SPECTRUM (POWER OF 2)'
      INPUT NPTS
      OUTPUT NPTS
C
C              CHECK IF IT'S APPROPRIATE TO ASK ABOUT WINDOWING
C              (IF IT IS, DO IT)
C
369   IF(ISW(1).OR.ISW(5).OR.ISW(6)) GO TO 391
      GO TO 392
391    OUTPUT 'WINDOWING FOR FOURIER TRANSFORMS (1/0)'
       INPUT IWNDW
      OUTPUT IWNDW
C
C              AND, LAST BUT NOT LEAST...
C
392   IF (ISW(1)) GO TO 393
      GO TO 394
393    OUTPUT 'FFT AVERAGING TIME CONSTANT'
      INPUT NAV
      OUTPUT NAV
394   IF (NAV.LE.0) NAV=1
      OUTPUT 'NUMBER OF ESTIMATES  '
      INPUT NEST
      OUTPUT NEST
      IF (NEST.GT.1000) NEST=1000
      IF (ISPCPLT .EQ. 1) GO TO 395
      GO TO 396
395   OUTPUT 'SPECTRUM PLOTS: LENEAR (0) OR LOG(1)'
      INPUT LOGSCL
      OUTPUT LOGSCL
396    IF(.NOT.ISW(6)) NDAT=64
C             OUTPUT'INPUT WGTS(1)'
C             INPUT WGTS(1)
C             OUTPUT'INPUT WGTS(2)'
C             INPUT WGTS(2)
C              OUTPUT'INPUT WGTS(3)'
C             INPUT WGTS(3)
C             OUTPUT'INPUT WGTS(4)'
```

```
C              INPUT WGTS(4)
      OUTPUT'DO YOU WANT HDCOPY 1-YES,0-NO'
       INPUT HDANS
       IF (HDANS.EQ.1)CALL HDCOPY
       CALL ERASE
        RETURN
        END
C
C
C
C
       SUBROUTINE INIT
C
C*********************************************************************
C
C                THE GRAND INITIALIZING ROUTINE
C
C
C*********************************************************************
C
      COMMON/SPEC/ ISPCFL(7),ISPC
      COMMON/INITIAL/ IRP,IFP,MTFOP
      COMMON NEST,NAV,IFST,IWNDW,LORD,ALPHA,WGTS(128),NO
      COMMON/SIGNL/ SIG(1024),RSLT(1024),WORK(1024),
     1NPTSF,NPTS,NSTRT,NTOT,NDAT,PSIG,SNRF,SNR,
     2ISYNDAT,GSIG,GNOISE,FREQ,SIGMA
      COMMON/PLS/ XNORM(10),ICT,LOGSCL,FIRST,IPXFL,LNAME(4),LPL
      COMMON/LBLS/LABEL(20,8)
      LOGICAL FIRST,LPL
      COMPLEX SIG,WGTS
      DIMENSION DATUMS(2048)
      DIMENSION IDAT(1024)
      EQUIVALENCE (DATUMS(1),SIG(1))
      DO 100 I=1,600
  100 WORK(I)=0.
C
C
C                SYNTHETIC DATA (NBRP) DOES'NT NEED ANYTHING DONE FOR IT
C
C                NORMALIZE MULTIPLICATION FACTORS FOR SNR
C                TOTAL INPUT POWER (VARIANCE) IS UNITY
C
  500 SNRF=10.**(SNR/10.)
      PNOISE=1./(1.+SNRF)
      PSIG=1.-PNOISE
      GNOISE=SQRT(PNOISE/2.)
      GSIG=SQRT(PSIG)
C
C                GET NUMBER OF TRAINING POINTS (NO) FOR ALPF
C
C     NO=NTOT-NDAT
C
```

```
C               GET PARAMETERS FOR INPUT-FFT
C
            NPTSF=NPTS
C           IF(NPTS.GE.NTOT) GO TO 600
C           M=ALOG(FLOAT(NTOT)-1.)/ALOG(2.)+1.
C           NPTSF=2**M
C
C               SET PARAMETERS FOR PLSPEC
C
  600       FIRST=.TRUE.
            IPXFL=0
C
            XNORM(1)=PSIG*NTOT + PNOISE
C
            XNORM(2)=(SNRF*LORD + 1.) ** 2
C
            XNORM(3)=XNORM(2)
C
            XNORM(4)=XNORM(2)
C
            XK=1./LORD + 1./SNRF
C           MAGN OF ALPF WEIGHT VECTOR
            PX=XK*XK
            XNORM(5)=PX*LORD
C
            PX=(XK*LORD)**2 * PSIG
C           ALPF GAIN**2 * PSIG
            XNORM(6)=PX*NDAT
C
            XNORM(7)=1.
            XNORM(8)=XNORM(1)
            XNORM(9)=1.
            XNORM(10)=1.
C
            RETURN
            END
C
C
C
            SUBROUTINE GETSIG (SIGPWR)
C
C           SUBROUTINE TO GET THE APPROPRIATE SIGNAL TO BE PROCESSED
C
C**********************************************************************
C
C
C               THERE ARE THREE DIFFERENT OUTPUTS OF SIGNALS:
C
C
C               SYNTHETIC/SINUSOIDAL--WITH RANDOM NOISE
C                         (ISYNDAT=1)
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C               SYNTHETIC/NBRP--NARROW-BAND RANDOM PROCESS
C               WITH RANDOM NOISE
C                         (ISYNDAT=2)
C
C               NSEED=0 => GENERATE NEW DATA
C
C
C*************************************************************************
C
C
C               COMMON BLOCK WITH SIGNAL PARAMETERS
C
C
           COMMON/SNR/RMS,SUMXI
      COMMON/PLS/XNORM(10),ICT,LOGSCL,IPXFL,LNAME(4),LPL
      COMMON NEST,NAV,IEST,IWNDW,LOPD,ALPHA,WGTS(128),NO
     COMMON /SIGNL/ SIG(1024),RSLT(1024),WORK(1024),
    1NPTSF,NPTS,NSTRT,NTOT,NDAT,PSIG,SNRF,SNR,
    2ISYNDAT,GSIG,GNOISE,FREQ,SIGMA
     COMPLEX SIG,WGTS,CSCAL
     INTEGER RBYTE
     DIMENSION DATUMS(2048),IDAT(1024),RAN(1)
     EQUIVALENCE (DATUMS(1),SIG(1))
      LOGICAL FIRST
       DATA NSEED/0/
      IF(NSEED.EQ.0)CALL RNDU(RAN,NSEED)
C
C               GO TO APPROPRIATE ROUTINE
C
     IS=ISYNDAT+1
     GO TO (2000,2000,3000) IS
C
C
C*************************************************************************
C
C                         SYNTHETIC DATA
C                         (SINUSOIDAL)
C
C               GET SINE WAVE FROM DISK
C                  (INIT HAS PLACED COSINE IN REAL PART, SINE IN
C                  IMAGINARY PART)
C
C               SYNTHETIC DATA (SINUSOIDAL) NEEDS A SINE WAVE IN 'SINE'
C               (REAL=COSINE, IMAGINARY=SINE)
C
 2000 OUTPUT ''
      IF (FREQ)401,402,402
  401 FREQ=FREQ+1.
  402 W=6.28318508*FREQ
     DO 410 I=1,NPTSF
     THETA=W*FLOAT(I-1)
     SIG(I)=CMPLX(COS(THETA),SIN(THETA))
```

```
C        IF(SIG(I).EQ.0.0)OUTPUT I,THETA
  410    CONTINUE
         TXO=2.0
         FIRST=.TRUE.
         SIGPWR=1.
         ONE=1.
         IST=1
          M=7
C
C                 ADD RANDOM NOISE TO EACH PART
C
         NTOT2=NPTSF*2
         DO 2300 I=1,NTOT2,2
          DATUMS(I)=DATUMS(I)*GSIG+DRAN(DUM)*GNOISE
 2300     DATUMS(I+1)=DATUMS(I+1)*GSIG+DRAN(DUM)*GNOISE
         SIGPWR=1.
         GO TO 4000
C
C*********************************************************************
C
C
C                            SYNTHETIC DATA
C                      (NARROW-BAND RANDOM PROCESS)
C
C
C                 GENERATE SIGNAL SPECTRUM AND SIGNAL POWER
C                 USING FREQUENCY AT MIDPOINT (.5)
C
 3000    PI=3.14159265359
         C=1./(SIGMA*SQRT(2.*PI))
         SIG2=2.*SIGMA*SIGMA
         PSIG=0.
         DO 3210 I=1,NPTSF
         DFREQ=FLOAT(I-1)/FLOAT(NPTSF)-.5
         RSLT(I)=0
         IF (ABS(DFREQ).LT.(5.*SIGMA)) RSLT(I)=C*EXP(-(DFREQ*DFREQ)/SIG2)
 3210    PSIG=PSIG+RSLT(I)
C
C                 CALCULATE NOISE POWER
C
         PNOISE=PSIG/SNRF
C
C                 NORMALIZE TO UNIT TOTAL POWER
C
         PSUM=PSIG+PNOISE
         PSIG=PSIG/PSUM
         PNOISE=PNOISE/PSUM
         SIGPWR = 1.0
C
C                 RANDOMIZE TO EXPONENTIAL DISTRIBUTION
C
            SCA=1./FLOAT(NPTSF)
```

```
      DO 3220 I=1,NPTSF
       NSEED=1
      CALL RNDU(RAN,NSEED)
 3220 RSLT(I)=-ALOG(RAN(1))*SCA*(PSIG*RSLT(I)+RNOISE)
C
C             GENERATE COMPLEX SPECTRUM FOR INPUT TO INVERSE FFT
C             SHIFTING SPECTUM OVER SO THAT CENTER FREQUENCY IS
C             OVER SPECIFIED FREQUENCY (FREQ) AND 'TAILS' OF
C             THE SPECTRUM ARE CYCLIC
C
 3221   CONTINUE
      FREQ2=FREQ
      IF (FREQ.LT.0.) FREQ2=FREQ+1.
      NL=IFIX(FREQ2*NPTSF)-NPTSF/2+NPTSF*2
          AVG=0.
      DO 3230 I=1,NPTSF
      ID=MOD((I+NL),NPTSF)+1
      INDXI=2*ID
      INDXR=INDXI-1
      XN=SQRT(RSLT(I))*.1
       NSEED=1
       CALL RNDU(RAN,NSEED)
      ARGUM=2.*PI*RAN(1)
      DATUMS(INDXR)=XN*COS(ARGUM)
 3230 DATUMS(INDXI)=XN*SIN(ARGUM)
          AVG=AVG/NPTSF
C            OUTPUT AVG
C
C             GENERATE COMPLEX TIME SERIES FROM INVERSE FFT
C
      M=ALOG(FLOAT(NPTSF))/ALOG(2.)+.5
        ONE1=-1.
      CALL FFT (M,ONE1)
        SCAL=FLOAT(NPTSF)
        CSCAL=CMPLX(SCAL,0.)
           SUMXI=0.
         DO 3233 NDAN=1,2*NPTSF
          SIG(NDAN)=CSCAL*SIG(NDAN)
 3233    SUMXI=CABS(SIG(NDAN))**2+SUMXI
         RMS=SQRT(SUMXI/(NPTSF*2))
         SCAL=1/RMS
         CSCAL=CMPLX(SCAL,0.)
          SUMXI=0.
           DO 3234 NDAN=1,2*NPTSF
           SIG(NDAN)=CSCAL*SIG(NDAN)
 3234     SUMXI=SUMXI+CABS(SIG(NDAN))**2
            RMS=SQRT(SUMXI/(NPTSF*2))
C        NPTSF=64
 4000 RETURN
      END
C
C
```

```
C
      SUBROUTINE TIMGRF (SIGPWR,SCL)
C
      COMMON/PLS/XNORM(10),ICT,LOGSCL,FIRST,IFXFL
      COMMON/SIGNL/SIG(1024),RSLT(1024),WORK(1024),
     1NPRTSF,NPTS,NSTRT,NTOT,NDAT,PSIG,SNRF,SNR,
     2ISYNDAT,CSIG,GNOISE,FREQ,SIGMA
      COMMON /SW/ ISW(0:15)
       COMMON/SMR/RMS,SUMX!
      DIMENSION IDT(3),ITM(3)
       COMPLEX SIG,WGTS
      LOGICAL FIRST, ISW
    1 XMIN = 0.
      XMAX = FLOAT(NPTS)
        YMIN=-SCL
        YMAX=SCL
C     IF (.NOT.ISW(0)) RETURN
             CALL ERASE
      DO 100 IC=1,2
C
C                 IF IC =1 PLOT REAL DATA. IF IC = 2 PLOT COMPLEX DATA
C
      CALL TWINDO(50,960,410,760)
      IF (IC.EQ.2) CALL TWINDO(50,960,60,410)
      CALL DWINDO (XMIN,XMAX,YMIN,YMAX)
C
C                 DRAW AXIS AND TIC MARKS
C
      CALL MOVEA (XMIN,YMIN)
      CALL DRAWA (XMIN,YMAX)
      CALL MOVEA (0.,0.)
      CALL DRAWA (XMAX,0.)
C
      CALL MOVEA (XMIN,YMIN)
      CALL SEELOC (IX,IY)
      CALL DRWABS (IX+30,IY)
C
C                 DRAW TICS
C
      NY = 10
      YY = 0.
      NYP1 = NY + 1
      YINC = (YMAX - 0.)/NY
      DO 20 J=1,NYP1
      CALL MOVEA (0.,YY)
      CALL SEELOC (IX,IY)
      CALL DRWABS (IX+15,IY)
   20 YY = YY + YINC
      YY = 0.
      YINC = (0. - YMIN)/NY
      DO 30 K=1,NYP1
      CALL MOVEA (0.,YY)
```

```
      CALL SETLOC (IX,IY)
      CALL DRWABS (IX+15,IY)
   30 YY = YY - YINC
C
      Y = REAL(SIG(1))
      IF (IC.EQ.2) Y = AIMAG(SIG(1))
      X = 0.
      CALL MOVEA (X,Y)
      DO 40 I=2,NPTS
      X = FLOAT (I-1)
      Y = REAL(SIG(I))
      IF (IC.EQ.2) Y=AIMAG(SIG(I))
   40 CALL DRAWA (X,Y)
  100 CONTINUE
C
C                LABEL GRAPH
C
      CALL  MOVABS (100,750)
      CALL ANMODE
      OUTPUT 'REAL'
      CALL RECOVR
      CALL MOVABS (100,400)
      CALL ANMODE
      OUTPUT 'IMAGINARY'
      CALL RECOVR
      CALL MOVABS (100,60)
      CALL ANMODE
      SIGRMS=SQRT(SIGPWR)
      OUTPUT 'RMS SIGNAL LEVEL = ',RMS,' NO. POINTS =    ',NPTS
        FIRST=.TRUE.
          ISW(0)=.FALSE.
       OUTPUT 'HARD COPY 1-YES,0-NO;'
       INPUT HDAN
       IF(HDAN.EQ.1)CALL HDCOPY
       CALL ERASE
      RETURN
      END
C
C
C
      SUBROUTINE SETFFT
C
C*************************************************************************
C
C                ROUTINE TO SET UP FOR FFT, WINDOW INPUT ARRAY, AND STORE
C                RESULT IN RSLT
C
C                SIG      INPUT SIGNAL
C                NPTS     NUMBER OF POINTS IN TRANSFORM
C                NDAT     NUMBER OF NON-ZERO DATA POINTS
C                IWNDW    WINDOWING FLAG (IF = 1= THEN WINDOW)
C                RSLT     ARRAY TO HOLD RESULT
```

```
C
C***********************************************************************
C
C

      COMMON/SIGNL/SIG(1024),RSLT(1024),WORK(1024),
     1NPTSF,NPTS,NSTRT,NTOT,NDAT,PSIG,SNRF,SNR,
     2ISYNDAT,GSIG,GNOISE,FREQ,SIGMA
      COMMON NEST,NAV,IEST,IWNDW,LORD,ALPHA,WCTS(126),NO
        COMMON/SW/ISW(0:15)
       LOGICAL FIRST,ISW
      COMPLEX SIG,WCTS
C
C                  FIGURE OUT 2**M=NPTS
C
      M=ALOG(FLOAT(NPTSF))/ALOG(2.)+.5
C
C                  DO WE WANT WINDOWING  ((IWNDW=1, YES)
C
       IWDO=IWNDW
       IF(ISW(8)) GOTO 1091
        GOTO 1092
 1091    IWDO=0
 1092  IF (IWDO.NE.1) GO TO 100
C
C                  YES
C
C                  FIGURE OUT INCREMENT FOR THETA BETWEEN EACH
C                  ELEMENT IN ARRAY
C
      WFAC=(2.*3.14159265359)/FLOAT(NDAT+1)
C
C                  PROCESS ENTIRE ARRAY THROUGH HANNING WINDOW
C                    (INCREMENTING ARGUMENT TO COSINE (THETA) BY
C                    WFAC EACH TIME)
C
      THETA=0.
      DO 10 I=1,NDAT
      THETA=THETA+WFAC
      SIG(I)=SIG(I)*.5*(1.-COS(THETA))
C      IF(SIG(I).EQ.0.0)OUTPUT SIG(I),THETA,WFAC
 10    CONTINUE
C
C                  FILL OUT WITH ZEROES IF NDAT<NPTS
C
 100  NDAT1=NDAT+1
      DO 110 I=NDAT1,NPTSF
 110  SIG(I)=(0.,0.)
C
C                  DO FFT ON SIGNAL
C
      ONE = 1.
```

```
      CALL FFT (M,ONE)
C
C           PLACE POWER SPECTRUM INTO RESULT
C
      DO 200 I=1,NPTSF
      RSLT(I)=CABS(SIG(I))**2/FLOAT(NDAT)
C     IF(RSLT(I).EQ.0.0) OUTPUT I,SIG(I),NDAT,RSLT(I)
  200 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE CALPF
C     ALPF (ADAPTIVE LINEAR PREDICTION FILTER)
C     ARRAY X IS USED TO UPDATE COEFFICIENT VECTOR G ACCORDING TO THE
C     LMS (NOISY GRADIENT) ADAPTIVE ALGORITHM
C     FILTER IS FILLED BEFORE ADAPTION BEGINS
C
         COMMON/LA/FLAGG
      COMMON/SIGNL/SIG(1024),RSLT(1024),WORK(1024),
     1NPTSF,NPTS,NSTRT,NTOT,NDAT,PSIG,SNRF,SNR,
     2ISYNDAT,CSIG,CNOISE,FREQ,SIGMA
      COMMON NFST,NAV,IFST,IWNDW,LORD,ALPHA,WCTS(128),NQ
      COMPLEX SIG ,WCTS ,XOUTT,ER,EROR
        XMU=ALPHA/FLOAT(LORD)
      DO 1 I=1,LORD
    1 WCTS(I)=0.
      LORDP=LORD+1
C
C           DO TRAINING
C
      DO 100 K=LORDP, 448
      XOUTT=(0.,0.)
C
C     FILTER TO GET KTH OUTPUT
C
      DO 10 L=1,LORD
      KK=K-L
      XOUTT=XOUTT+WCTS(L)*SIG(KK)
   10 CONTINUE
C
C     ADAPT WEIGHTS
C
      ER=SIG(K)-XOUTT
      EROR=ER*XMU
      DO 20 L=1,LORD
      KK=K-L
      WCTS(L)=WCTS(L)+EROR*CONJG(SIG(KK))
   20 CONTINUE
  100 CONTINUE
```

```
  101 NOP=NO+1
C
C               TRAINING OVER, SAVE OUTPUT
C
      DO 200 K=449,512
      XOUTT=(0.,0.)
C
C     FILTER TO GET KTH OUTPUT
C
      DO 210 L=1,LORD
      KK=K-L
      XOUTT=XOUTT+WGTS(L)*SIG(KK)
  210 CONTINUE
C
C     ADAPT WEIGHTS
C
      ER=SIG(K)-XOUTT
      EROR=ER*XMU
      DO 220 L=1,LORD
      KK=K-L
      WGTS(L)=WGTS(L)+EROR*CONJG(SIG(KK))
C       OUTPUT L,WGTS(L)
  220 CONTINUE
C     EMS=EMS+CABS(ER)
      SIG(K-448)=XOUTT
  200 CONTINUE
C     IF (NTOT.NE.NO) EMS=EMS/(NTOT-NO)
      RETURN
      END
```

```
C
C
C
      SUBROUTINE CCORR
C
C**********************************************************************
C
C                 COMPLEX CORRELATION ROUTINE
C
C
C                 SIG       SIGNAL
C                 LORD      ORDER
C                 R         OUTPUT COEFFICIENTS
C                 NTOT      NUMBER OF TOTAL DATA POINTS IN SIG
C
C**********************************************************************
C
      COMMON/SW/ISW(0:15)
      COMMON NEST,NAV,IEST,IWNDW,LORD,ALPHA,WGTS(128),NQ
      COMMON/SIGNL/ SIG(1024),RSLT(1024),WORK(1024),
     1NPTSF,NPTS,NSTRT,NTOT,NDAT,PSIG,SNRF,SNR,
     2ISYNDAT,GSIG,GNOISE,FREQ,SIGMA
      COMPLEX SIG,R(129),CSUM,WGTS
      EQUIVALENCE (R(1),RSLT(1))
        LOGICAL ISW
       ILORD=LORD
C
C    IF SIG ARRAY IS COMMING FROM ALPF USE THE FIRST NDAT PTS OF THE AR
C
C        IF (.NOT.(ISW(12))) GOTO 1093
C     NOD=1
C     NDO=NDAT
C       GOTO 1096
CC
CC    IF SIG ARRAY IS NOT COMMING FROM ALPF USE THE LAST NDAT PTS
CC
 1093  NOD= 1
        NDO=NDAT
 1096  DO 100 L=0,ILORD
      CSUM=CMPLX(0.,0.)
      NML=NDO-L
      DO 200 K=NOD,NML
  200 CSUM=CSUM+CONJG(SIG(K))*SIG(K+L)
  100 R(L+1)=CSUM/NTOT
      RETURN
      END
```

```
      SUBROUTINE SPKPKP (IST,J)
C
C
C**********************************************************************
C
C
C              ROUTINE TO CALCULATE FIRST CIRCULAR MOMENT (MEAN
C              FREQUENCY) AND HALF POWER WIDTH OVER ENTIRE SPECTRUM
C              AND THEN RECALCULATE POWER OVER ONLY SPECTRUM
C              BANDWIDTH ABOUT MEAN FREQ.
C
C
C              RSLT    ARRAY WITH SPECTRUM
C              IST     NUMBER OF ESTIMATOR
C              J       NUMBER OF ESTIMATION
C              NPTS    NUMBER OF POINTS IN RSLT
C
C**********************************************************************
C
C
       COMMON/PKS/PEAKS(3,1024,8)
      COMMON/SIGNL/SIG(1024),RSLT(1024),WORK(1024),
     1NPTSF,NPTS,NSTRT,NTOT,NDAT,PSIG,SNRF,SNR
      COMMON/PLS/XNORM(10),ICT,LOGSCL,FIRST,IPXFL
       COMPLEX SIG
C
C     FIND FIRST CIRCULAR MOMENT AND WIDTH
C
      DELW=6.28319/FLOAT(NPTSF)
      SUMC=0.0
      SUMS=0.0
      POWER = 0.0
      FRQM1=0.0
      WIDTH = 0.0
      DO 100 I=1,NPTSF
      SI=RSLT(I)
      W=DELW*FLOAT(I-1)
      SUMC=SUMC+SI*COS(W)
      SUMS=SUMS+SI*SIN(W)
      POWER = POWER + SI
      FRQM1 = FRQM1 + SI*W
      WIDTH = WIDTH + SI*W*W
  100 CONTINUE
      FRQM1=FRQM1/POWER/6.28319
      FREQDA=ATAN2(SUMS,SUMC)/6.28319
C      OUTPUT J,FREQDA
      NFRQ=FREQDA*NPTSF+0.5
C
C     (2*PI)**2 = 39.4784
C
      WIDTH=WIDTH/POWER/39.4784 - FRQM1*FRQM1
      WIDTH=SQRT(WIDTH)
```

```
      IF(FREQDA.GT.0.5) FREQDA=FREQDA-1.0
C
C     SUM UP POWER OVER SPECTRUM WIDTH
C
      NP=2.*WIDTH*NPTSF
      N1=NFRQ-NP
      NN1=NPTSF+N1
      IF(N1.LT.1) N1=1
      N2=NFRQ+NP
      IF(N2.GT.NPTSF) N2=NPTSF
      POWER=0.
      DO 200 I=N1,N2
200   POWER=POWER+RSLT(I)
      DO 201 I=NN1,NPTSF
201   POWER=POWER+RSLT(I)
      POWER=POWER/XNORM(IST)
C
C     PUT THE THREE VARIABLES INTO AN ARRAY, AND
C
        PEAKS(1,J,IST)=POWER
        PEAKS(2,J,IST)=FREQDA
        PEAKS(3,J,IST)=WIDTH
C         OUTPUT J,FREQDA
      RETURN
      END
C
C
C
      SUBROUTINE PLSPEC ( IST)
C
C*************************************************************************
C
C
C                  ROUTINE TO PLOT THE SPECTRUM IN RSLT
C
C                  RSLT     ARRAY CONTAINING SPECTRUM
C                  NPTS     NUMBER OF POINTS IN RSLT
C                  IST      NUMBER OF ESTIMATOR
C                  IREC     NUMBER OF RECORD
C                  LOGSCL   FLAG FOR LINEAR OR LOG PLOT
C                  XNORM    ARRAY CONTAINING NORMALIZATIONS
C
C*************************************************************************
C
      COMMON/SIGNL/SIG(1024),RSLT(1024),WORK(1024),
     1NPTSF,NPTS,NSTRT,NTOT,NDAT,PSIG,SNRF,SNR,
     2ISYNDAT,GSIG,GNOISE,FREQ,SIGMA
      COMMON/PLS/ XNORM(10),ICT,LOGSCL,FIRST,IPXFL,LNAME(4),LPL
      COMMON NEST,NAV,IEST,IWNDW,LORD,ALPHA,WGTS(128),NO
        COMMON /SW/ISW(0:15)
       COMPLEX SIG,WGTS
      LOGICAL FIRST,IPL,PLCH,ISW
```

```
      DIMENSION  IDT(3),ITM(3)
        ISTT=IST
 1       IF (ISW(12)) GOTO 95
          GOTO 97
 95        IF (IST.EQ.6) ISTT=8
 97   IF (LOGSCL) 100,100,200
C
C                 LINEAR SCALE
C
100   YMIN=0.
      YMAX=XNORM(ISTT)
      YTIC=YMAX/10.
      GO TO 300
C
C                 LOG SCALE
C
200   XNRM=XNORM(ISTT)
         YMIN=-80.
         YMAX=3.
C     YMAX=3.
      YTIC=10.
C
C                 SET UP WINDOW
C
300   CALL TWINDO (50,1000,25,700)
C        NPTSF=2*NPTSF
      XMAX=FLOAT(NPTSF/2)
      XMIN=1.-XMAX
      CALL DWINDO (XMIN,XMAX,YMIN,YMAX)
C
C                 IF SWITCH 13 IS UP, DO AN AXIS AND RESET IPXFL
C
 360   CONTINUE
      IF ((.NOT.FIRST)) GO TO 400
         AX1=1.
         AX2=YMIN
         AX3=FLOAT(NPTSF)/10.
         AX4=YTIC
        CALL AXIS(AX1,AX2,AX3,AX4)
      IPXFL=0
C
C                 DETERMINE OUTPUT OF DASHED LINE
C
400   L=ISTT-1
      IF (ISTT.EQ.8) L=3
      IF (ISTT.EQ.6) L=4
      IF (ISTT.EQ.5) L=0
C
C                 PLOT SPECTRUM
C
      NPTS2=NPTSF/2
      Y=RSLT(NPTS2+1)
```

```
      IF(Y.EQ.0)OUTPUT Y,XNRM
      IF(LOGSCL.GT.0) Y=10.*ALOG10(Y/XNRM)
      CALL MOVEA (XMIN,Y)
      IMIN=2-NPTS2
      DO 500 I=IMIN,0
      X=I
      Y=RSLT(NPTSF+I)
      IF(LOGSCL.GT.0) Y=10.*ALOG10(Y/XNRM)
      CALL DASHA (X,Y,L)
500   CONTINUE
      DO 501 I=1,NPTS2
      X=I
      Y=RSLT(I)
      IF(LOGSCL.GT.0) Y=10.*ALOG10(Y/XNRM)
      CALL DASHA (X,Y,L)
501   CONTINUE
600   CALL ANMODE
C       NPTSF=NPTSF/2
      RETURN
      END
C
C
C
      SUBROUTINE SPCAV (ISTT,ILP10)
C
C     BLOCK AVERAGE FIRST NAV ESTIMATES
C     EXPONENTIAL AVERAGE AFTER FIRST NAV
C     PUT IN RSLT FOR PLOTTING
C
      COMMON/SIGNL/ SIG(1024),RSLT(1024),WORK(1024),
     1NPTSF,NPTS,NSTRT,NTOT,NDAT,PSIG,SNRF,SNR,
     2ISYNDAT,GSIG,GNOISE,FREQ,SIGMG
      COMMON NEST,NAV,IEST,IWNDW,LORD,ALPHA,WGTS(128),NO
        COMPLEX SIG,WGTS
      FNAV=FLOAT(NAV)
      IF(ILP10.LE.NAV) FNAV=FLOAT(ILP10)
      XD=1./FNAV
      XA=1.-XD
      DO 30 I=1,NPTSF
      WORK(I)=XA*WORK(I)    + XD*RSLT(I)
30    RSLT(I) =WORK(I)
      RETURN
      END
C
C
C
      SUBROUTINE DSKRD (IST,IHIST)
C
C
C*****************************************************************
C
C
```

```
C
C      ROUTINE TO READ PEAKS,POWERS,AND VARIANCES FROM PEAKS
C
C
C****************************************************************************
C
C
       COMMON NEST,NAV,IEST,IWNDW,LORD,ALPHA,WGTS(128),NO
         COMMON/PKS/PEAKS(3,1024,8)
        COMMON/SIGNL/ SIG(1024),RSLT(1024),WORK(1024),
      1NPTSF,NPTS,NSTRT,NTOT,NDAT,PSIG,SNRF,SNR,
      2ISYNDAT,GSIG,GNOISE,FREQ,SIGMA
         COMPLEX SIG,WGTS
C
C                 STORE REQUESTED INFO IN WORK
C
       DO 100 I=1,NEST
          WORK(I)=PEAKS(IHIST,I,IST)
  100     CONTINUE
       RETURN
       END
C
C
C
       SUBROUTINE STATS ( XMEAN,STDEV,NBINS)
C
C                 ROUTINE TO DO STATISTICAL ANALYSES ON WORK (NEST)
C
C                 OUTPUTS FROM STATS
C                   XMEAN--> MEAN OF W
C                   STDEV--> STANDARD DEVIATION OF W
C
       COMMON/SIGNL/ SIG(1024),RSLT(1024),WORK(1024),
      1NPTSF,NPTS,NSTRT,NTOT,NDAT,PSIG,SNRF,SNR,
      2ISYNDAT,GSIG,GNOISE,FREQ,SIGMA
       COMMON NEST,NAV,IEST,IWNDW,LORD,ALPHA,WGTS(128),NO
          COMPLEX SIG,WGTS
       SUM=0
       SUMSD=0
          DW=6.28319/FLOAT(NBINS)
            SUMC=0.
            SUMS=0.
       DO 10 I=1,NBINS
        W=RSLT(I)
        XW=DW*FLOAT(I-1)
         SUMC=SUMC+W*COS(XW)
         SUMS=SUMS+W*SIN(XW)
  10   CONTINUE
        XMEAN=ATAN2(SUMS,SUMC)
C           OUTPUT XMEAN
          IF(XMEAN.GT.0) GOTO 91
           XMEAN=-1*XMEAN
```

```
             XMEAN=6.28319-XMEAN
C               OUTPUT XMEAN
 91        DO 97 I=1,NBINS
             XW=DW*FLOAT(I-1)
           XMD=XMEAN-XW
             IF(XMD.LT.0)INDEX=I-1
           IF(XMD.LT.0)GOTO 98
 97          CONTINUE
C            OUTPUT INDEX,DW
 98            INDEX=INDEX-NBINS/2-1
               XMEAN=(INDEX-.5)/FLOAT(NBINS)
C            OUTPUT XMEAN,INDEX
          DO 19 I=1,NEST
          ATEST=WORK(I)-XMEAN
          ATEST=ATEST**2
          TEST=SQRT(ATEST)
          TEST1=1.-TEST
          IF(TEST.GT..5)ATEST=TEST1**2
 19         SUMSD=ATEST+SUMSD
            STDEV=SQRT(SUMSD/(NEST-1))
         RETURN
         END
C
C
C
         SUBROUTINE EHIST (HMAX,NBINS,IHIST)
C
C                    ROUTINE TO DO HISTOGRAM ON WORK(NEST)
C                      WORK--> INPUT ARRAY WITH RAW DATA
C                      RSLT--> OUTPUT ARRAY WITH HISTOGRAM
C                      HMAX--> LARGEST VALUE IN HIST
C
         COMMON/SIGNL/ SIG(1024),RSLT(1024),WORK(1024),
        1NFTSF,NPTS,NSTRT,NTOT,NDAT,PSIG,SNRF,SNR,
        2ISYNDAT,GSIG,SNOISE,FREQ,SIGMG
         COMMON NEST,NAV,IEST,IWNDW,LORD,ALPHA,WGTS(128),NO
           COMPLEX SIG,WGTS
         HMAX=1.
         DO 10 I=1,NBINS
10       RSLT(I)=0.
         DO 100 I=1,NEST
         IF (IHIST-2) 200,300,400
C
C        POWER NORMALIZED TO (0,2.0)
200      INDEX=0.5*FLOAT(NBINS)*WORK(I)+0.5
         GO TO 500
C
C        FREQUENCY NORMALIZED TO (0-.5,+0.5)
C
300      INDEX=FLOAT(NBINS)*WORK(I)+0.5
         INDEX=INDEX + NBINS/2 + 1
         GO TO 500
```

```
C
C     WIDTH NORMALIZED TO (0.0,0.5)
C
400   INDEX=2.*FLOAT(NBINS)*WORK(I)+0.5
500   IF (INDEX.LT.1) INDEX=1
      IF(INDEX.GT.NBINS) INDEX=NBINS
      RSLT(INDEX)=RSLT(INDEX) + 1
      IF(RSLT(INDEX).GT.HMAX) HMAX=RSLT(INDEX)
          IF(RSLT(INDEX).EQ.HMAX) HIND=INDEX
100   CONTINUE
C        OUTPUT HMAX,HIND
      RETURN
      END
C
C
C
      SUBROUTINE HPLOT (HMAX,XMEAN,STDEV,NBINS,IHIST,IST)
C
C**********************************************************************
C
C
C                  ROUTINE TO PLOT HISTOGRAMS FROM RSLT
C
C
C**********************************************************************
C
C
      COMMON NEST,NAV,IEST,IWNDW,LORD,ALPHA,WGTS(128),NO
      COMMON/SIGNL/ SIG(1024),RSLT(1024),WORK(1024),
     1NPTSF,NPTS,NSTRT,NTOT,NDAT,PSIG,SNRF,SNF,
     2ISYNDAT,GSIG,GNOISE,FREQ,SIGMA
      COMMON/PLS/ XNORM(10),ICT,LOGSCL,FIRST,IPXFL
      COMMON/LBLS/LABEL(10,8)
      COMPLEX SIG,WGTS
      DIMENSION IDT(3),ITM(3)
      IPC=(HMAX/FLOAT(NEST))*100.+1.
      PC=FLOAT(IPC)*FLOAT(NEST)/100.
      CALL DWINDO (1.,FLOAT(NBINS),0.,PC)
      CALL TWINDO (100,1000,100,750)
      PC10=PC/10.
      PTS10=FLOAT(NBINS)/10.
      XAX=1.
      IF (IHIST.EQ.2) XAX=NBINS/2+1
      CALL  AXIS (XAX,0.,PTS10,PC10)
      CALL RECOVR
      CALL MOVEA (1.,0.)
      DO 10 I=1,NBINS
      CALL DRAWA (FLOAT(I),RSLT(I))
      CALL DRAWA (FLOAT(I+1),RSLT(I))
10    CONTINUE
      CALL HOME
      CALL ANMODE
```

```
X        ACCEPT '<7>',IDUM
 110     WRITE (10,1090) LORD,NBINS
1090     FORMAT ('  (ORDER=',I3,' BINS=',I4,')')
C        GO TO 200
C130     WRITE (10,1030) LORD,ALPHA,NBINS
1030     FORMAT (' (ORDER=',I3,' ALP=',F4.2,' BINS=',I4,')')
 200     WRITE (10,2000) NEST,NTOT,SNR,XMEAN,STDEV
2000     FORMAT (' ',I4,' ESTIMATES ',I4,' SAMPLES/ESTIMATE ',F5.1,
        1  ' DB SNR',/,' MEAN:',F8.5,' ST. DEV.:',F7.5,Z)
             OUTPUT ' '
             OUTPUT ' '
           IF(IST.EQ.1)OUTPUT 'FFT HISTOGRAM'
           IF(IST.EQ.5)OUTPUT 'WGTS HISTOGRAM'
           IF(IST.EQ.6)OUTPUT 'ALPF HISTOGRAM'
           IF(IST.EQ.8)OUTPUT  'MUTILAG HISTOGRAM'
              OUTPUT ' '
              OUTPUT ' '
           OUTPUT'HARD COPY 1-YES,0-NO'
           INPUT HDAN
            IF(HDAN.EQ.1)CALL HDCOPY
         CALL NEWPAG
         RETURN
         END
C
C
C
         SUBROUTINE  FFT (M,TNV)

C
C        DESCRIPTION OF PARAMETERS:
C        SIG-AS INPUT CONTAINS COMPLEX ONE-DIMENSIONAL
C        ARRAY TO BE TRANSFORMED
C         -AS OUTPUT-CONTAINS THE COMPLEX FOURIER TRANSFORM
C        M-DEFINES THE DIMENSION OF A.
C        THE SIZE OF N(OF A) = 2*:M
C        N-DIMENSION OF A
C        TNV-OPTION PARAMETER
C              1. O TAKES TRANSFORM OF A
C             -1. O TAKES INVERSE TRANSFORM OF A
C         NOTE:  N MUST BE A POWER OF 2
C
         COMMON/SIGNL/ SIG(1024),RSLT(1024),WORK(1024),
        1NPTSF,NPTS,NSTRT,NTOT,NDAT,PSIG,SNRF,SNF,
        2ISYNDAT,GSIG,GNOISE,FREQ,SIGMA
         COMPLEX SIG,U,W,T,C
           N=2**M
C           DO 2 IID=1,N
C2          OUTPUT SIG(IID)
         NV2=N/2
         NM1=N-1
         J=1
C              OUTPUT N,NV2,J,TNV
```

```
C
C     THIS LOOP PERFORMS THE BIT-REVERSAL
C     OPERATION ON THE INDEX OF A
C
      DO 7 I=1,NM1
      IF (I .GE. J) GO TO 5
      T=SIG(J)
      SIG(J)=SIG(I)
      SIG(I)=T
5     K=NV2
6     IF (K .GE. J) GO TO 7
      J=J-K
      K=K/2
      GO TO 6
7        J=J+K
      PI=3.14159265359
      TNV=- TNV
      DO 20 L=1,M
      LE=2 **L
      LE1=LE/2
      U=(1.,0.)
      W=CMPLX(COS(PI/LE1),TNV*SIN(PI/LE1))
      DO 20 J=1,LE1
      DO 10 I=J,N,LE
      IP=I+LE1
      T=SIG(IP)*U
      SIG(IP)=SIG(I)-T
10    SIG(I)=SIG(I)+T
20    U=U*W
       TNV=-TNV
C         OUTPUT N,NV2,J,TNV
C         DO 21 IDID=1,N
C21       OUTPUT SIG(IDID)
      IF (TNV .GT. 0) RETURN
C
C     IF TNV=-1 WE HAVE THE INVERSE TRANSFORM
C
      S=1./N
      C=CMPLX(S,0.)
      DO 30 I=1,N
       SIG(I)=C*SIG(I)
C       OUTPUT SIG(I)
 30       CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE AXIS(X,Y,XTIC,YTIC)
      CALL SEEDW(XMIN,XMAX,YMIN,YMAX)
      CALL MOVEA(XMIN,Y)
      CALL DRAWA(XMAX,Y)
```

```
      CALL MOVEA(X,YMIN)
      CALL DRAWA(X,YMAX)
      IF (XTIC .LE. 0) GO TO 10
      NX=(XMAX-XMIN)/XTIC+1
      MNX=(X-XMIN)/XTIC
      X0=X-XTIC*MNX
      DO 20 I=1,NX
      CALL MOVEA(X0,Y)
      CALL MOVREL(0,10)
      CALL DRWREL(0,-20)
      X0=X0+XTIC
20    CONTINUE
10    IF (YTIC .LE. 0) GO TO 30
      NY = (YMAX-YMIN)/YTIC+1
      MNY=(Y-YMIN)/YTIC
      Y0=Y-YTIC*MNY
      DO 40 I=1,NY
      CALL MOVEA (X,Y0)
      CALL MOVREL (10,0)
      CALL DRWREL (-20,0)
      Y0=Y0+YTIC
40    CONTINUE
30    CALL HOME
      RETURN
      END
      FUNCTION IBITR(J,NU)
      J1=J
      IBITR=0
      DO 200 I=1,NU
      J2=J1/2
      IBITR=IBITR*2+(J1-2*J2)
  200 J1=J2
      RETURN
      END
C
C
C
        FUNCTION DRAN(DUM)
C
C          RANDOM + - # BETWEEN 0 + 1
C
         DIMENSION RAN(1)
       NSEED=1
       CALL RNDU(RAN,NSEED)
         NSS=100*RAN(1)
        CALL RNDU(RAN,NSEED)
          DS=-1**(NSS*RAN(1))
       CALL RNDU(RAN,NSEED)
          DRAN=DS*RAN(1)
           RETURN
           END
```